

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

**THIS PAGE BLANK (USPTO)**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: )  
COFLER )  
Serial No. 10/083,629 )  
Filing Date: February 26, 2002 )  
For: METHOD OF HANDLING INSTRUCTIONS )  
WITHIN A PROCESSOR WITH )  
DECOUPLED ARCHITECTURE, IN )  
PARTICULAR A PROCESSOR FOR )  
DIGITAL SIGNAL PROCESSING, AND )  
CORRESPONDING PROCESSOR )

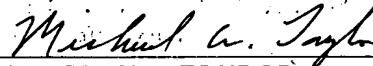
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Director, U.S. Patent and Trademark Office  
Washington, D.C. 20231

Sir:

Transmitted herewith is a certified copy of the  
priority French Application No. 0102647.

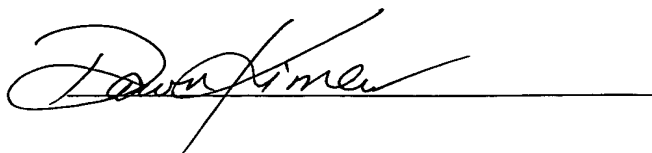
Respectfully submitted,



MICHAEL W. TAYLOR  
Reg. No. 43,182  
Allen, Dyer, Doppelt, Milbrath  
& Gilchrist, P.A.  
255 S. Orange Avenue, Suite 1401  
Post Office Box 3791  
Orlando, Florida 32802  
Telephone: 407/841-2330  
Fax: 407/841-2343  
Attorney for Applicant

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being  
deposited with the United States Postal Service as first class  
mail in an envelope addressed to: DIRECTOR, U.S. PATENT AND  
TRADEMARK OFFICE, WASHINGTON, D.C. 20231, on this 14<sup>th</sup> day of  
March, 2002.



**THIS PAGE BLANK (USPTO)**



# BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 22 FEV. 2002

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

Martine PLANCHE

**THIS PAGE BLANK (USPTO)**



26 bis, rue de Saint Pétersbourg  
75800 Paris Cedex 08  
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

# BREVET D'INVENTION CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI



REQUÊTE EN DÉLIVRANCE 1/2

Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 W / 260899

<b>REMISE DES PIÈCES</b> DATE <b>27 FEV 2001</b> LIEU <b>75 INPI PARIS</b> N° D'ENREGISTREMENT <b>0102647</b> NATIONAL ATTRIBUÉ PAR L'INPI DATE DE DÉPÔT ATTRIBUÉE <b>27 FEV. 2001</b> PAR L'INPI		<b>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE</b> À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE BUREAU D.A. CASALONGA-JOSSE 8, Avenue Percier 75008 PARIS	
<b>Vos références pour ce dossier</b> (facultatif) B 00/4116 FR FZ			
<b>Confirmation d'un dépôt par télécopie</b> <input type="checkbox"/> N° attribué par l'INPI à la télécopie			
<b>2 NATURE DE LA DEMANDE</b>		<b>Cochez l'une des 4 cases suivantes</b>	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
<i>Demande de brevet initiale</i> <i>ou demande de certificat d'utilité initiale</i>		N° _____ Date ____/____/____ N° _____ Date ____/____/____	
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>		<input type="checkbox"/> N° _____ Date ____/____/____	
<b>3 TITRE DE L'INVENTION</b> (200 caractères ou espaces maximum) Procédé de gestion d'instructions au sein d'un processeur à architecture découplée, en particulier un processeur de traitement numérique du signal, et processeur correspondant.			
<b>4 DÉCLARATION DE PRIORITÉ</b> <b>OU REQUÊTE DU BÉNÉFICE DE</b> <b>LA DATE DE DÉPÔT D'UNE</b> <b>DEMANDE ANTÉRIEURE FRANÇAISE</b>		Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suit »	
<b>5 DEMANDEUR</b>		<input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»	
Nom ou dénomination sociale		STMicroelectronics SA	
Prénoms			
Forme juridique		Société Anonyme	
N° SIREN			
Code APE-NAF			
Adresse	Rue	7 avenue Galliéni	
	Code postal et ville	94250	GENTILLY
Pays		FRANCE	
Nationalité		Française	
N° de téléphone (facultatif)			
N° de télécopie (facultatif)			
Adresse électronique (facultatif)			



# BREVET D'INVENTION CERTIFICAT D'UTILITÉ

REQUÊTE EN DÉLIVRANCE 2/2

REMISE DES PIÈCES	
DATE	<b>27 FEV 2001</b>
LIEU	<b>75 INPI PARIS</b>
N° D'ENREGISTREMENT	<b>0102647</b>
NATIONAL ATTRIBUÉ PAR L'INPI	

D9 540 W / 190600

<b>Vos références pour ce dossier :</b> (facultatif)		B 00/4116 FR
<b>6 MANDATAIRE</b> Nom Prénom Cabinet ou Société N° de pouvoir permanent et/ou de lien contractuel Adresse Rue Code postal et ville N° de téléphone (facultatif) N° de télécopie (facultatif) Adresse électronique (facultatif)		BUREAU D.A. CASALONGA-JOSSE 8, Avenue Percier 75008 PARIS
<b>7 INVENTEUR (S)</b>		
Les inventeurs sont les demandeurs		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Dans ce cas fournir une désignation d'inventeur(s) séparée
<b>8 RAPPORT DE RECHERCHE</b>		Uniquement pour une demande de brevet (y compris division et transf. rmination)
Établissement immédiat ou établissement différé		<input checked="" type="checkbox"/> Établissement immédiat <input type="checkbox"/> Établissement différé
Paiement échelonné de la redevance		Paiement en deux versements, uniquement pour les personnes physiques <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non
<b>9 RÉDUCTION DU TAUX DES REDEVANCES</b>		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Requête antérieurement à ce dépôt (joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence)
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes		
<b>10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE</b> (Nom et qualité du signataire)		<b>VISA DE LA PRÉFECTURE OU DE L'INPI</b>  A. CASALONGA (bm 92-10441) Conseil en Propriété Industrielle



**Procédé de gestion d'instructions au sein d'un processeur à architecture découplée, en particulier un processeur de traitement numérique du signal, et processeur correspondant.**

L'invention concerne les processeurs, notamment les processeurs de traitement numérique du signal (DSP : "Digital Signal Processors" en langue anglaise), et notamment la gestion d'instructions au sein d'un tel processeur, telles que les instructions de chargement dans un registre d'une donnée mémorisée, en particulier lorsqu'elles sont gardées.

Un processeur concerné par la présente invention comporte généralement plusieurs unités de traitement fonctionnant en parallèle. Classiquement, il est prévu en tant qu'unités de traitement, une unité arithmétique et logique, une unité d'adressage et une unité de gestion de branchement. Outre ces unités de traitement, le processeur comporte généralement une unité de contrôle ou unité centrale qui dialogue avec la mémoire-programme et délivre aux différentes unités de traitement, des instructions élémentaires encore communément appelées micro-instructions.

Par ailleurs, le processeur concerné par la présente invention est à architecture découplée, permettant, après un temps de latence initiale, une vitesse d'exécution des instructions plus importante.

Le principe d'une architecture découplée est déjà connu de l'homme du métier, par exemple par la demande de brevet européen n° 0 949 565 ou par la demande de brevet européen n° 1 050 799. On rappelle maintenant brièvement le principe d'une architecture découplée, l'homme du métier pouvant se référer pour plus de détails, si nécessaire, aux demandes de brevets européens précitées.

Outre les unités de traitement précitées, il est prévu une interface mémoire qui contient notamment une mémoire du type FIFO (premier entré-premier sorti) destinée à recevoir et à stocker temporairement les données contenues dans une mémoire de données, par exemple formée de deux bancs de mémoire SRAM.

Dans une architecture découplée, une instruction de chargement dans un registre d'une donnée mémorisée est partitionnée en deux micro-

instructions ou instructions. Une première instruction est transmise au module d'adressage (unité d'adressage) qui calcule l'adresse effective de la donnée dans la mémoire de données. L'autre instruction, qui est une instruction de chargement dans le registre concerné, est temporairement stockée dans une mémoire FIFO associée à l'unité arithmétique et logique. Cette deuxième instruction reste en attente jusqu'à ce que la donnée mémorisée, élaborée par l'unité d'adressage, soit disponible. Lorsqu'il en est ainsi, la mise à jour du registre concerné est effectivement réalisée. Au bout d'un temps de latence initiale, l'unité d'adressage a pris

10 de l'avance sur l'unité arithmétique et logique. La machine est alors "découplée". En contrepartie, vu de l'unité arithmétique et logique, la latence "fictive" est nulle.

Cependant, puisque les instructions destinées à l'unité arithmétique et logique sont traitées temporellement dans l'ordre, car elles sont stockées dans une FIFO, il est tout à fait possible qu'une première instruction de chargement d'une donnée mémorisée dans un premier registre soit à un instant donné stockée en tête de la FIFO, et par conséquent prête à être délivrée aux étages de "pipeline" de l'unité de traitement, et qu'une deuxième instruction impliquant des registres 20 différents du premier registre, soit stockée juste derrière cette première instruction de chargement. Et, tant que cette première instruction de chargement reste bloquée en tête de la FIFO en attente de la donnée mémorisée élaborée par l'unité d'adressage, la deuxième instruction, immédiatement derrière celle-ci, reste également bloquée alors qu'elle est totalement indépendante de ladite instruction bloquée en tête de la FIFO.

L'invention vise à apporter une solution à ce problème.

L'invention a pour but de proposer un mécanisme de gestion des instructions totalement différent de celui existant jusqu'à maintenant, et qui puisse améliorer les performances générales d'un processeur à 30 architecture découplée.

L'invention propose donc un procédé de gestion d'instructions au sein d'un processeur à architecture découplée, ce processeur comportant un cœur contenant plusieurs unités de traitement respectivement associées à des moyens de mémoire du type FIFO (premier entré-premier sorti) pour stocker séquentiellement les

instructions respectives qui sont destinées aux unités correspondantes. Par ailleurs, la délivrance au coeur du processeur, d'une instruction de chargement dans un registre d'une donnée mémorisée, provoque la délivrance au moyen de mémoire d'une première unité de traitement, par exemple une unité arithmétique et logique, d'une instruction de chargement dans le registre et la délivrance au moyen de mémoire d'une deuxième unité de traitement, par exemple l'unité d'adressage, d'une instruction destinée à élaborer ladite donnée mémorisée. L'instruction de chargement n'est exécutée que lorsque ladite donnée mémorisée a été délivrée par ladite deuxième unité de traitement.

Selon une caractéristique générale de l'invention, le moyen de mémoire de la première unité de traitement comporte une première mémoire de type FIFO, et une deuxième mémoire de type FIFO distincte de la première. On stocke chaque instruction de chargement dans la première mémoire et certaines au moins des autres instructions opératives destinées à la première unité dans la deuxième mémoire (en pratique, on verra ci-après que selon un mode de mise en oeuvre particulièrement avantageux, il est préférable en présence d'instructions gardées, dont la signification sera détaillée ci-après, de stocker des instructions dites de transmission dans une troisième mémoire). On extrait de la deuxième mémoire une instruction opérative impliquant au moins un registre et ayant atteint la tête de cette deuxième mémoire, si aucune instruction de chargement temporellement plus ancienne et destinée à modifier la valeur du ou des registres associée à cette instruction opérative, n'est présente dans la première mémoire. Et, en présence d'une telle instruction modificatrice de chargement temporellement plus ancienne, on extrait ladite instruction opérative de la deuxième mémoire seulement après que l'instruction modificatrice de chargement a été extraite de la première mémoire.

En d'autres termes, selon l'invention, une instruction de chargement dans un registre d'une donnée mémorisée ne présente plus un "caractère bloquant" vis-à-vis d'instructions opératives temporellement plus jeunes qui n'impliquent pas au moins un registre concerné par cette instruction de chargement.

Selon un mode de mise en oeuvre du procédé selon l'invention, à

à chaque fois qu'on stocke dans la deuxième mémoire une instruction sans qu'il soit prévu de stocker simultanément dans la première mémoire une instruction de chargement, on stocke dans la première mémoire une instruction non-opérative (instruction NOP, selon une dénomination bien connue de l'homme du métier). Par ailleurs, à chaque fois qu'on extrait une instruction de la première mémoire, quelle qu'elle soit, on incrémente un premier compteur de lecture. A chaque fois qu'on stocke une instruction, quelle qu'elle soit, dans la première mémoire, on incrémente un premier compteur d'écriture.

Et, à chaque fois qu'on stocke dans la première mémoire une instruction de chargement, on mémorise la valeur courante du premier compteur d'écriture. Et, la détermination du caractère toujours présent de cette instruction de chargement dans la première mémoire prend en compte le résultat de la comparaison de ladite valeur courante mémorisée

du premier compteur d'écriture avec la valeur courante du premier compteur de lecture.

En d'autres termes, on détermine si une instruction de chargement qui a été écrite dans la première mémoire est toujours présente dans cette première mémoire, en utilisant la comparaison de la valeur courante mémorisée du premier compteur d'écriture avec la valeur courante mémorisée du premier compteur de lecture.

En pratique, le premier compteur de lecture et le premier compteur d'écriture ont avantageusement une taille binaire identique et égale à la profondeur de la première mémoire. On associe alors, selon un

mode de mise en oeuvre du procédé, à chaque premier compteur, un bit de dépassement changeant de valeur à chaque fois que le premier compteur correspondant revient à sa valeur initiale. Et, à chaque fois qu'on stocke dans la première mémoire une instruction de chargement, on mémorise également la valeur courante du bit de dépassement du premier compteur

d'écriture. Et, la détermination du caractère toujours présent de cette instruction de chargement dans la première mémoire prend également en compte le résultat de la comparaison de la valeur courante du bit de dépassement du premier compteur de lecture avec ladite valeur mémorisée du bit de dépassement du premier compteur d'écriture.

Ceci permet de tenir compte du fait que le pointeur de lecture et

le pointeur d'écriture de la première mémoire peuvent se situer dans deux fenêtrés "circulaires" différentes.

En ce qui concerne les instructions opératives impliquant au moins un registre, selon un mode de mise en oeuvre du procédé selon l'invention, à chaque fois qu'on stocke dans la deuxième mémoire une instruction opérative impliquant au moins un registre, on lui associe une première "étiquette" contenant la valeur courante du premier compteur d'écriture qui a été mémorisée lorsque la dernière (c'est à dire la plus jeune) instruction de chargement modifiant l'un au moins des registres impliqués dans ladite instruction opérative a été stockée dans la première mémoire. Le critère d'extraction de cette instruction opérative ayant atteint la tête de la deuxième mémoire prend alors en compte le résultat de la comparaison entre ladite valeur courante mémorisée associée à cette instruction opérative (contenue dans la première étiquette), et la valeur courante du premier compteur de lecture.

En d'autres termes, le principe de base est ici de comparer la valeur courante du premier compteur de lecture avec la valeur courante du premier compteur d'écriture qui a été mémorisée lorsque la dernière instruction de chargement modifiant l'un au moins des registres impliqués dans ladite instruction opérative a été stockée dans la première mémoire.

Et, le principe de base consiste à autoriser l'extraction de l'instruction opérative ayant atteint la tête de la mémoire lorsque la valeur courante du premier compteur de lecture est supérieure à la valeur courante mémorisée contenue dans la première étiquette qui a été attachée à cette instruction opérative.

Cependant, là encore, il convient de prendre en compte notamment le cas où une instruction de chargement est écrite alors que le premier compteur d'écriture est dans une fenêtré circulaire donnée et que le premier compteur de lecture est toujours positionné dans la fenêtré circulaire précédente.

Aussi, pour résoudre de tels cas particuliers, selon un mode de mise en oeuvre du procédé selon l'invention, à chaque fois qu'on stocke une instruction opérative dans la deuxième mémoire, on lui associe également dans sa première étiquette la valeur courante du bit de dépassement du premier compteur d'écriture qui a été mémorisée lorsque

la dernière instruction de chargement modifiant l'un au moins des registres impliqués dans ladite instruction opérative a été stockée dans la première mémoire. Et, le critère d'extraction de cette instruction opérative ayant atteint la tête de la deuxième mémoire prend également en compte le résultat de la comparaison entre cette valeur courante mémorisée associée à cette instruction opérative et la valeur courante du bit de dépassement du premier compteur de lecture.

Parmi les instructions délivrées aux unités de traitement, figurent des instructions dites "gardées" permettant une écriture plus performante du code programme.

L'utilisation d'instructions gardées dans un processeur est déjà connue de l'homme du métier, par exemple par la demande de brevet européen n° 1 050 803. On rappelle ici brièvement le principe des instructions gardées, l'homme du métier pouvant se référer pour plus de détails, si nécessaire, à la demande de brevet européen précitée.

Une instruction gardée est une instruction affectée d'une indication dite "de garde", en pratique un bit de garde pouvant prendre la valeur 0 ou 1. L'instruction gardée sera alors exécutée ou non en fonction de la valeur du bit de garde, c'est-à-dire de l'indication de garde. Toutes les valeurs binaires des indications de garde, pouvant être au nombre de seize par exemple, sont contenues dans un registre dit "registre d'indications de garde". Ce registre centralisé est incorporé au sein d'une unité de traitement, par exemple l'unité arithmétique et logique, et contient à chaque instant les valeurs courantes des indications ou bits de garde. Les valeurs de ces indications de garde peuvent être modifiées par des instructions spécifiques dites "instructions modificatrices".

Pour lire la valeur d'une indication de garde associée à une instruction gardée, l'unité de traitement qui reçoit cette instruction gardée et qui n'a pas un accès direct au registre des indications de garde (par exemple l'unité de branchement), requiert un transfert de la valeur de l'indication de garde depuis l'unité de traitement qui détient le registre des indications de garde. En d'autres termes, la délivrance d'une instruction gardée à une unité de traitement qui ne détient pas le registre des indications de garde, provoque la délivrance à l'unité de traitement qui est la gardienne du registre des indications de garde, d'une instruction dite

"de transmission" qui est destinée à faire transmettre à l'unité de traitement qui reçoit l'instruction gardée, la valeur de l'indication de garde associée à cette instruction gardée.

Et, l'instruction gardée est maintenue dans l'étage de tête (étage de sortie) de la mémoire d'instructions du type FIFO, associée à cette unité de traitement, jusqu'à ce que cette dernière reçoive de l'unité de traitement qui détient le registre des indications de garde, ladite valeur de l'indication de garde associée à cette instruction gardée. Ce n'est que lorsque l'unité de traitement qui détient le registre des indications de garde, aura exécuté l'instruction de transmission, que la valeur correspondante de l'indication de garde sera transmise à l'unité de traitement requérante, de façon que celle-ci puisse ou non exécuter son instruction gardée.

Or, si pour une raison ou pour une autre, l'instruction de transmission est bloquée dans la mémoire d'instructions de l'unité de traitement qui détient le registre des indications de garde, l'unité de traitement requérante va être en conséquence également bloquée jusqu'à ce qu'elle reçoive la valeur de l'indication de garde correspondante.

L'invention vise également à apporter une solution à ce problème.

L'invention a également pour but de proposer un mécanisme de gestion des instructions gardées totalement différent de celui existant jusqu'à maintenant, et qui puisse améliorer les performances générales d'un processeur utilisant un registre centralisé des indications de garde.

Par ailleurs, les instructions de chargement peuvent être elles-mêmes des instructions gardées. Or, compte tenu de la présence de la première mémoire, qui est destinée à stocker les instructions de chargement, il est nécessaire de s'assurer que la valeur d'une indication de garde délivrée par le registre des indications de garde et destinée à une instruction de chargement, est bien la bonne valeur attendue compte tenu de l'exécution dans l'ordre de toutes les instructions stockées dans les différentes FIFO.

L'invention apporte également une solution à ce problème.

Plus précisément, selon un mode de mise en oeuvre de l'invention, le moyen de mémoire de la première unité comporte une

troisième mémoire de type FIFO, distincte de la première et de la deuxième mémoire, ainsi qu'une quatrième mémoire de type FIFO distincte des trois premières. La première unité contient par ailleurs un registre d'indications de garde. La délivrance d'une instruction gardée au moyen de mémoire d'une autre unité de traitement différente de la première, par exemple l'unité de branchement, ou la délivrance d'une instruction de chargement gardée à la première unité, provoque la délivrance à la première unité de traitement et son stockage dans la troisième mémoire, d'une instruction de transmission destinée à faire transmettre à ladite autre unité (par exemple l'unité de branchement) ou à ladite quatrième mémoire, respectivement, la valeur de l'indication de garde associée à ladite instruction gardée ou à ladite instruction de chargement gardée, respectivement.

On extrait alors de la troisième mémoire, une instruction de transmission ayant atteint la tête de cette troisième mémoire, si aucune instruction modificatrice temporellement plus ancienne destinée à modifier la valeur de l'indication de garde associée à cette instruction de transmission, n'est présente dans la deuxième mémoire. Et, en présence d'une telle instruction modificatrice temporellement plus ancienne dans la deuxième mémoire, on extrait l'instruction de transmission de la troisième mémoire seulement après que l'instruction modificatrice a été extraite de la deuxième mémoire.

En d'autres termes, selon l'invention, une instruction de transmission peut "court-circuiter" toutes les autres instructions si aucune instruction temporellement plus ancienne stockée dans la deuxième mémoire n'est de nature à modifier la valeur de l'indication de garde associée à cette instruction de transmission.

Par ailleurs, selon l'invention, la quatrième mémoire qui est destinée à recevoir les valeurs des indications de garde des instructions de chargement gardées, est lue en parallèle avec la première mémoire destinée à recevoir les instructions de chargement, évitant ainsi une perte de synchronisation dans l'exécution des instructions.

Selon un mode de mise en oeuvre du procédé selon l'invention, à chaque fois qu'on extrait une instruction de la deuxième mémoire, on incrémente un deuxième compteur de lecture. De même, à chaque fois



qu'on stocke une instruction dans la deuxième mémoire, on incrémente un deuxième compteur d'écriture.

Par ailleurs, à chaque fois qu'on stocke dans la deuxième mémoire une instruction modifiant la valeur d'une indication de garde, on mémorise la valeur courante du deuxième compteur d'écriture. Et, la détermination du caractère toujours présent d'une instruction modificatrice dans la deuxième mémoire prend en compte le résultat de la comparaison de ladite valeur courante mémorisée du deuxième compteur d'écriture avec la valeur courante du deuxième compteur de lecture.

10 En d'autres termes, on détermine si une instruction modificatrice qui a été écrite dans la deuxième mémoire est toujours présente dans cette deuxième mémoire, en utilisant la comparaison de la valeur courante mémorisée du deuxième compteur d'écriture avec la valeur courante du deuxième compteur de lecture.

15 En pratique, le deuxième compteur de lecture et le deuxième compteur d'écriture ont avantageusement une taille binaire identique égale à la profondeur de la deuxième mémoire. On associe alors, selon un mode de mise en oeuvre du procédé, à chaque deuxième compteur, un bit de dépassement changeant de valeur à chaque fois que le deuxième

20 compteur correspondant revient à sa valeur initiale. Et, à chaque fois qu'on stocke dans la deuxième mémoire une instruction modifiant la valeur d'une indication de garde, on mémorise également la valeur courante du bit de dépassement du deuxième compteur d'écriture. Et, afin de tenir compte du fait que le pointeur de lecture et le pointeur d'écriture de la deuxième mémoire peuvent se situer dans deux fenêtres "circulaires" 25 différentes, la détermination du caractère toujours présent de l'instruction modificatrice dans la deuxième mémoire prend également en compte le résultat de la comparaison de la valeur courante du bit de dépassement du deuxième compteur de lecture avec ladite valeur mémorisée du bit de dépassement du deuxième compteur d'écriture.

30 En ce qui concerne les instructions de transmission, selon un mode de mise en oeuvre du procédé selon l'invention, à chaque fois qu'on stocke une instruction de transmission dans la troisième mémoire et que l'on ne stocke pas simultanément une autre instruction dans la deuxième 35 mémoire, on stocke simultanément dans la deuxième mémoire une

instruction non-opérative (instruction NOP) de façon à ne pas désynchroniser les mécanismes. Par ailleurs, à chaque stockage d'une instruction de transmission, on associe à ladite instruction de transmission une deuxième "étiquette" comportant la valeur courante du deuxième compteur d'écriture qui a été mémorisée lorsque la dernière instruction modifiant l'indication de garde associée à ladite instruction de transmission a été stockée dans la deuxième mémoire. Le critère d'extraction de cette instruction de transmission ayant atteint la tête de la troisième mémoire, prend alors en compte le résultat de la comparaison entre ladite valeur courante mémorisée associée à cette instruction de transmission (contenue dans la deuxième étiquette), et la valeur courante du deuxième compteur de lecture.

En d'autres termes, le principe de base est ici de comparer la valeur courante du deuxième compteur de lecture avec la valeur courante du deuxième compteur d'écriture qui a été mémorisée lorsque la dernière instruction modifiant l'indication de garde associée à cette instruction de transmission a été stockée dans la deuxième mémoire. Et, le principe de base consiste à autoriser l'extraction de l'instruction de transmission ayant atteint la tête de la mémoire lorsque la valeur courante du deuxième compteur de lecture est supérieure à la valeur courante mémorisée contenue dans l'étiquette qui a été attachée à cette instruction de transmission.

Cependant, là encore, il convient de prendre en compte notamment le cas où une instruction modificatrice est écrite alors que le deuxième compteur d'écriture est dans une fenêtre circulaire donnée et que le deuxième compteur de lecture est toujours positionné dans la fenêtre circulaire précédente.

Aussi, pour résoudre de tels cas particuliers, selon un mode de mise en oeuvre du procédé selon l'invention, à chaque fois qu'on stocke une instruction de transmission dans la troisième mémoire, on lui associe également dans sa deuxième étiquette, la valeur courante du bit de dépassement du deuxième compteur d'écriture qui a été mémorisée lorsque la dernière instruction modifiant l'indication de garde associée à ladite instruction de transmission a été stockée dans la deuxième mémoire. Et, le critère d'extraction prend alors également en compte le

résultat de la comparaison entre cette valeur courante mémorisée du bit de dépassement du deuxième compteur d'écriture, contenue dans la deuxième étiquette attachée à cette instruction de transmission, et la valeur courante du bit de dépassement du deuxième compteur de lecture.

5 L'invention a également pour objet un processeur à architecture découplée, comprenant un coeur contenant plusieurs unités de traitement, respectivement associées à des moyens de mémoire du type FIFO pour stocker séquentiellement les instructions respectives qui sont destinées aux unités correspondantes. Le coeur de processeur comporte par ailleurs

10 une unité centrale apte en réponse à une instruction de chargement dans un registre d'une donnée mémorisée, à délivrer au moyen de mémoire d'une première unité de traitement, une instruction de chargement dans ledit registre, et à délivrer au moyen de mémoire d'une deuxième unité de traitement une instruction destinée à élaborer ladite donnée mémorisée.

15 L'instruction de chargement n'est exécutée que lorsque ladite donnée mémorisée a été délivrée par ladite deuxième unité de traitement.

Selon une caractéristique générale de l'invention, le moyen de mémoire de la première unité comporte une première mémoire de type FIFO, et une deuxième mémoire de type FIFO distincte de la première.

20 Le processeur comporte par ailleurs :  
- des moyens d'aiguillage ("dispatcher" en langue anglaise) aptes à stocker chaque instruction de chargement dans la première mémoire et certaines au moins des autres instructions opératives destinées à la première unité de traitement dans la deuxième mémoire, et

25 des premiers moyens de commande aptes à extraire de la deuxième mémoire une instruction opérative impliquant au moins un registre et ayant atteint la tête de cette deuxième mémoire si aucune instruction de chargement temporellement plus ancienne et destinée à modifier la valeur du ou des registres associés à cette instruction opérative, n'est présente dans la première mémoire, et  
30 en présence d'une telle instruction modificatrice de chargement temporellement plus ancienne, à extraire l'instruction opérative de la deuxième mémoire seulement après que l'instruction modificatrice de chargement a été extraite de la première mémoire.

35 Selon un mode de réalisation de l'invention, les premiers moyens

de commande comportent :

- un premier compteur de lecture incrémenté à chaque fois qu'une instruction est extraite de la première mémoire,
- un premier compteur d'écriture incrémenté à chaque fois qu'une instruction est stockée dans la première mémoire,
- un ensemble de premiers registres élémentaires respectivement associés à l'ensemble des registres capables d'être chargés par des données mémorisées,
- un premier bloc de commande apte, à chaque fois qu'une instruction de chargement dans un registre est stockée dans la première mémoire, à stocker la valeur courante du premier compteur d'écriture dans un champ principal du premier registre élémentaire associé à ce registre,
- un deuxième bloc de commande apte à déterminer le caractère toujours présent de cette instruction de chargement dans la première mémoire; ce deuxième bloc de commande comportant des premiers moyens de comparaison aptes à comparer le contenu dudit champ principal du premier registre élémentaire, avec la valeur courante du premier compteur de lecture.

Selon un mode de réalisation de l'invention, le premier compteur d'écriture et le premier compteur de lecture ont une taille binaire identique égale à la profondeur de la première mémoire. A chaque premier compteur, est associé un bit de dépassement changeant de valeur à chaque fois que le premier compteur correspondant revient à sa valeur initiale. Chaque premier registre élémentaire comporte en outre un champ auxiliaire à un bit. Le premier bloc de commande est alors apte, à chaque fois qu'une instruction de chargement dans un registre est stockée dans la première mémoire, à stocker également la valeur courante du bit de dépassement du premier compteur d'écriture dans le champ auxiliaire du premier registre élémentaire correspondant. Le deuxième bloc de commande comporte alors des premiers moyens de comparaison auxiliaires aptes à comparer la valeur courante du bit de dépassement du premier compteur de lecture avec le contenu du champ auxiliaire. Ces premiers moyens de comparaison auxiliaires comportent par exemple une porte logique NON OU EXCLUSIF (XNOR en langue anglaise).

Selon un mode de réalisation de l'invention, chaque étage de la deuxième mémoire comporte un champ utile pour le stockage d'une instruction opérative, et un premier champ supplémentaire. (Ce premier champ supplémentaire va permettre de stocker une partie de la première étiquette attachée à l'instruction opérative, typiquement la valeur mémorisée du premier compteur d'écriture).

Les premiers moyens de commande comportent alors :

- un troisième bloc de commande apte, à chaque fois qu'une instruction opérative impliquant au moins un registre est stockée dans le champ utile de l'étage d'entrée de la deuxième mémoire, à sélectionner parmi tous les premiers registres élémentaires associés à tous les registres impliqués dans ladite instruction opérative, celui correspondant à la dernière (la plus jeune) instruction de chargement modifiant l'un au moins de ces registres, et à transférer le contenu du champ principal du premier registre élémentaire sélectionné, dans le premier champ supplémentaire de l'étage d'entrée de la deuxième mémoire, et

- un quatrième bloc de commande apte à élaborer le critère d'extraction de cette instruction opérative ayant atteint l'étage de tête de la deuxième mémoire, ce quatrième bloc de commande comportant des premiers moyens de comparaison aptes à comparer le contenu du premier champ supplémentaire de l'étage de tête avec la valeur courante du premier compteur de lecture.

Selon un mode de réalisation de l'invention, chaque étage de la deuxième mémoire comporte en outre un deuxième champ supplémentaire. Ce deuxième champ supplémentaire, par exemple un bit supplémentaire, va permettre de stocker une autre partie de la première étiquette attachée à l'instruction opérative, typiquement la valeur mémorisée du bit de dépassement du premier compteur d'écriture. Le troisième bloc de commande est alors apte, à chaque fois qu'une instruction opérative est stockée dans le champ utile de l'étage d'entrée de la deuxième mémoire, à transférer le contenu du champ auxiliaire du premier registre élémentaire sélectionné, dans le deuxième champ supplémentaire de l'étage d'entrée de la deuxième mémoire. Le quatrième bloc de commande comporte alors des premiers moyens de comparaison supplémentaires aptes à comparer le contenu du deuxième champ

supplémentaire avec la valeur courante du bit de dépassement du premier compteur de lecture.

Là encore, ces moyens de comparaison supplémentaires peuvent comporter une porte logique NON OU EXCLUSIF (XNOR).

Selon un mode de réalisation de l'invention, plus particulièrement adapté à la gestion des instructions gardées, le moyen de mémoire de la première unité comporte une troisième mémoire de type FIFO, distincte de la première et de la deuxième mémoires, et une quatrième mémoire de type FIFO, distincte des trois premières.

La première unité de traitement contient un registre d'indications de garde.

L'unité centrale est alors apte à délivrer une instruction gardée au moyen de mémoire d'une autre unité de traitement différente de la première, ou bien une instruction de chargement gardée à la première unité de traitement, et

à délivrer au moyen de mémoire de la première unité de traitement, une instruction de transmission destinée à faire transmettre à ladite autre unité ou à ladite quatrième mémoire, respectivement, la valeur de l'indication de garde associée à ladite instruction gardée ou à ladite instruction de chargement gardée, respectivement.

Par ailleurs, les moyens d'aiguillage sont aptes à stocker chaque instruction de transmission dans la troisième mémoire.

Et le processeur comporte en outre des deuxième moyens de commande aptes :

à extraire de la troisième mémoire une instruction de transmission ayant atteint la tête de cette troisième mémoire si aucune instruction modificatrice temporellement plus ancienne et destinée à modifier la valeur de l'indication de garde associée à cette instruction de transmission, n'est présente dans la deuxième mémoire, et

en présence d'une telle instruction modificatrice temporellement plus ancienne, à extraire l'instruction de transmission de la troisième mémoire seulement après que l'instruction modificatrice a été extraite de la deuxième mémoire.

Selon un mode de réalisation de l'invention, les deuxième moyens de commande comportent :

un deuxième compteur de lecture incrémenté à chaque fois qu'une instruction est extraite de la deuxième mémoire,

un deuxième compteur d'écriture incrémenté à chaque fois qu'une instruction est stockée dans la deuxième mémoire,

5. un ensemble de deuxièmes registres élémentaires respectivement associés à l'ensemble des indications de garde,

un cinquième bloc de commande apte, à chaque fois qu'une instruction modifiant la valeur d'une indication de garde est stockée dans la deuxième mémoire, à stocker la valeur courante du deuxième compteur d'écriture dans un champ principal du deuxième registre élémentaire associé à cette indication de garde,

10. un sixième bloc de commande apte à déterminer le caractère toujours présent de cette instruction modificatrice dans la deuxième mémoire, ce sixième bloc de commande comportant des deuxièmes moyens de comparaison aptes à comparer le contenu dudit champ principal du deuxième registre élémentaire, avec la valeur courante du deuxième compteur de lecture.

Selon un mode de réalisation de l'invention, le deuxième compteur d'écriture et le deuxième compteur de lecture ont une taille binaire identique égale à la profondeur de la deuxième mémoire. A chaque deuxième compteur, est associé un bit de dépassement changeant de valeur à chaque fois que le deuxième compteur correspondant revient à sa valeur initiale. Chaque deuxième registre élémentaire comporte en outre un champ auxiliaire à un bit.

25. Le cinquième bloc de commande est alors apte, à chaque fois qu'une instruction modifiant la valeur d'une indication de garde est stockée dans la deuxième mémoire, à stocker également la valeur courante du bit de dépassement du deuxième compteur d'écriture dans le champ auxiliaire du deuxième registre élémentaire correspondant.

30. Le sixième bloc de commande comporte alors des deuxièmes moyens de comparaison auxiliaires aptes à comparer la valeur courante du bit de dépassement du deuxième compteur de lecture avec le contenu du champ auxiliaire. Ces deuxièmes moyens de comparaison auxiliaires comportent par exemple une porte logique NON OU EXCLUSIF (XNOR en langue anglaise).

35

Selon un mode de réalisation de l'invention, chaque étage de la troisième mémoire comporte un champ utile pour le stockage d'une instruction de transmission, et un premier champ supplémentaire. (Ce premier champ supplémentaire va permettre de stocker une partie de la deuxième étiquette attachée à l'instruction de transmission, typiquement la valeur mémorisée du deuxième compteur d'écriture).

Les deuxièmes moyens de commande comportent alors :

- un septième bloc de commande apte, à chaque fois qu'une instruction de transmission est stockée dans le champ utile de l'étage d'entrée de la troisième mémoire, à transférer le contenu du champ principal du deuxième registre élémentaire associé à l'indication de garde correspondante, dans le premier champ supplémentaire de l'étage d'entrée de la troisième mémoire; et

- un huitième bloc de commande apte à élaborer le critère d'extraction de cette instruction de transmission ayant atteint l'étage de tête de la troisième mémoire; ce huitième bloc de commande comportant des deuxièmes moyens de comparaison aptes à comparer le contenu du premier champ supplémentaire de l'étage de tête avec la valeur courante du deuxième compteur de lecture.

20 Selon un mode de réalisation de l'invention, chaque étage de la troisième mémoire comporte en outre un deuxième champ supplémentaire. Ce deuxième champ supplémentaire, par exemple un bit supplémentaire, va permettre de stocker une autre partie de la deuxième étiquette attachée à l'instruction de transmission, typiquement la valeur 25 mémorisée du bit de dépassement du deuxième compteur d'écriture. Le septième bloc de commande est alors apte, à chaque fois qu'une instruction de transmission est stockée dans le champ utile de l'étage d'entrée de la troisième mémoire, à transférer le contenu du champ auxiliaire du deuxième registre élémentaire associé à l'indication de garde correspondante, dans le deuxième champ supplémentaire de l'étage d'entrée de la troisième mémoire. Le huitième bloc de commande 30 comporte alors des deuxièmes moyens de comparaison supplémentaires aptes à comparer le contenu du deuxième champ supplémentaire avec la valeur courante du bit de dépassement du deuxième compteur de lecture. 35 Là encore, ces deuxièmes moyens de comparaison



supplémentaires peuvent comporter une porte logique NON OU EXCLUSIF (XNOR). D'autres avantages et caractéristiques de l'invention apparaîtront à l'examen de la description détaillée de modes de mise en oeuvre et de réalisation, nullement limitatifs, et des dessins annexés, sur lesquels :

la figure 1 illustre schématiquement l'architecture générale d'un processeur selon l'invention;

la figure 2 illustre plus en détail, mais toujours schématiquement le coeur du processeur de la figure 1;

la figure 3 illustre plus en détail, mais toujours schématiquement une architecture permettant une gestion des instructions de chargement dans un registre de données mémorisées, selon l'invention;

les figures 4 à 7 illustrent schématiquement des organigrammes relatifs à des modes de mise en oeuvre du procédé de gestion d'instructions de chargement de données mémorisées, selon l'invention;

la figure 8 illustre plus en détail, mais toujours schématiquement un mode de réalisation permettant en outre une gestion des instructions gardées, selon l'invention;

les figures 9 à 13 illustrent schématiquement des organigrammes relatifs à des modes de mise en oeuvre de la gestion d'instructions gardées, selon l'invention; et

la figure 14 illustre toujours schématiquement et plus spécifiquement les quatre mémoires FIFO présentes dans l'unité de traitement arithmétique et logique, ainsi que leur interdépendance temporelle, en association avec le registre des indications de garde.

Sur la figure 1, la référence PROC. désigne un processeur, par exemple un processeur de traitement numérique du signal.

Le processeur PROC. comporte une mémoire-programme PM contenant les instructions de programme. Cette mémoire-programme est adressée par un contrôleur de mémoire PMC qui coopère avec le coeur CR du processeur.

Le coeur CR du processeur dialogue également avec un contrôleur de mémoire DMC qui contrôle l'accès à une mémoire de

données DMM, par exemple deux bancs de mémoire SRAM.

Sur la figure 2, on voit que le coeur CR du processeur comporte une unité centrale CU et plusieurs unités de traitement, ici trois unités de traitement distinctes DU, AU et GU.

Dans le mode de réalisation illustré sur la figure 2, on ne prend pas en compte le fait que les instructions puissent être gardées. On ne décrira donc, en référence aux figures 2 à 7, uniquement que la gestion des instructions de chargement dans un registre des données mémorisées.

L'unité de contrôle CU est l'unité maîtresse du système. En général, elle gère toutes les activités de commande. Plus particulièrement, elle délivre au contrôleur de mémoire PMC des adresses programme, et répartit les instructions correspondantes vers les différentes unités de traitement. Plus précisément, l'unité centrale CU comporte une interface PFAL avec le contrôleur de mémoire PMC, qui effectue notamment un formatage des instructions en instructions prêtes à être décodées. Les instructions sont décodées ensuite dans un décodeur DCD qui transforme chaque instruction machine en une ou plusieurs micro-instructions qui pourront être traitées par les différentes unités de traitement. Dans la suite du texte, à des fins de simplification, on désigne sous le vocable "d'instruction" une micro-instruction.

Enfin, un moyen d'aiguillage DISP ("Dispatcher" en langue anglaise) envoie les instructions générées par le décodeur aux unités de traitement auxquelles sont destinées ces instructions.

Les unités de traitement du coeur de processeur comportent ici une unité arithmétique et logique DU qui exécute la plupart des opérations arithmétiques.

L'unité DU contient tous les registres Rx dans lesquels on souhaite stocker (charger) des données mémorisées. Ces registres Rx peuvent être au nombre de 16, par exemple.

Il est par ailleurs prévu une unité d'adressage AU qui effectue les calculs d'adresses effectives nécessaires pour adresser les opérandes dans la mémoire. Cette unité AU contient tous les registres de pointage Px nécessaires à cette tâche.

Une troisième unité de traitement est une unité de branchement GU qui effectue tous les calculs de branchement d'adresse et les transfère

à l'unité centrale CU.

L'unité de traitement DU comporte un moyen de mémoire du type FIFO composé, comme on le verra plus en détail ci-après, d'une première mémoire RLDQ destinée à stocker des instructions de chargement et une deuxième mémoire DIDQ destinée à stocker toutes les autres instructions destinées à cette unité DU, en particulier des instructions opératives DUOPx impliquant au moins un registre Rx.

Outre ces unités DU, AU et GU, il est prévu une interface mémoire DMI qui contient notamment une autre mémoire du type FIFO, LDQ, destinée à recevoir et à stocker temporairement les données contenues dans les bancs de mémoire SRAM.

Par ailleurs, au même titre que l'unité DU comporte un moyen de mémoire du type FIFO, chaque autre unité de traitement, à savoir l'unité AU et l'unité GU, contient également un moyen de mémoire du type FIFO, référencé AIDQ, et destiné à stocker les instructions destinées à ces unités de traitement.

On se réfère maintenant plus particulièrement à la figure 3, pour décrire le mécanisme selon l'invention de gestion des instructions de chargement de donnée mémorisée dans un registre.

A titre d'exemple, une instruction de chargement dans un registre Rx d'une donnée mémorisée s'écrit :

$LD\ Rx; [Py]$

Cette notation représente une instruction signifiant que l'on doit charger (stocker) dans le registre Rx la donnée mémorisée dans la mémoire DMM à l'adresse fournie par le contenu du registre de pointage Py.

L'unité centrale qui reçoit une telle instruction la scinde en une instruction d'élaboration de la donnée mémorisée, destinée à l'unité d'adressage AU, et en une instruction de chargement destinée à l'unité DU.

L'instruction d'élaboration va permettre à l'unité AU, via le contrôleur de mémoire DMC, de lire la donnée mémorisée à l'adresse fournie par le pointeur Py et de transférer cette donnée mémorisée à la mémoire LDQ.

L'instruction de chargement LDRx, qui s'écrit  $Rx=LDQ$ , va permettre à l'unité DU de charger (stocker) dans le registre Rx le contenu



de la mémoire LDQ.

Une instruction opérative DUOPx impliquant un registre Rx est par exemple une instruction du type  $Rx = Rx + 1$ , consistant dans ce cas à incrémenter d'une unité le contenu du registre Rx.

Une instruction opérative peut impliquer plus d'un registre, par exemple 4 registres. Une telle instruction opérative est alors par exemple  $RO = R_1 + R_2 + R_3$ .

Dans la suite du texte, la référence DUOPx désigne une instruction opérative, et x désigne le numéro du registre Rx impliqué dans cette instruction opérative.

Comme indiqué ci-avant, le moyen de mémoire associé à l'unité DU comporte une première mémoire du type FIFO, référencée RLDQ, et une deuxième mémoire du type FIFO, référencée DIDQ. La deuxième mémoire DIDQ est destinée à stocker toutes les instructions opératives INST destinées à l'unité DU, et notamment des instructions DUOPx.

Cette deuxième mémoire DIDQ comporte plusieurs étages (ici quatre), référencés LETG1-LETG4. Chaque étage LETGi comporte un champ utile LCHUi pour le stockage d'une instruction INST. L'écriture et

la lecture dans la deuxième mémoire DIDQ sont gérées par un pointeur d'écriture PE2 et un pointeur de lecture PL2. Une instruction INST qui aurait été stockée dans l'étage LETG1, va alors se propager séquentiellement depuis l'étage d'entrée LETG1 de la mémoire jusqu'à l'étage de sortie LETG4, ou étage de tête, de la mémoire DIDQ. Bien

entendu si à un instant donné, la FIFO est partiellement remplie ou vide, une instruction pourra être stockée directement au premier étage disponible, par exemple LETG2, ou LETG3 ou bien LETG4, comme cela est bien connu de l'homme du métier. Ainsi au sens de la présente invention l'étage d'entrée n'est donc pas nécessairement l'étage LETG1.

Chaque instruction de chargement LDRx impliquant un registre Rx est stockée dans la première mémoire RLDQ. L'écriture et la lecture dans cette première mémoire RLDQ sont gérées également de façon classique par un pointeur d'écriture PE1 et un pointeur de lecture PL1. Les deux mémoires RLDQ et DIDQ ont, dans l'exemple décrit ici, la même profondeur (taille) égale à quatre. Il convient également de noter que les pointeurs PE1 et PL1 associés à la mémoire RLDQ sont décorrélés (c'est-

à-dire commandés de façon indépendante) par rapport aux pointeurs PE2 et PL2 associés à la mémoire DIDQ.

Outre ces deux mémoires de type FIFO, le processeur comporte des premiers moyens de commande MCTL1, par exemple incorporés au sein de l'unité DU. Ces premiers moyens de commande vont permettre, comme on va le voir plus en détail ci-après, de délivrer dans certains cas les instructions opératives DUOp<sub>x</sub> beaucoup plus rapidement aux étages d'exécution pipelinés de l'unité DU.

Plus précisément, ces premiers moyens de commande MCTL1 vont extraire de la mémoire DIDQ une instruction opérative DUOp<sub>x</sub> ayant atteint la tête LETG4 de cette mémoire DIDQ, si aucune instruction de chargement LDR<sub>x</sub> temporellement plus ancienne destinée à modifier la valeur du registre Rx associé à cette instruction opérative DUOp<sub>x</sub>, n'est présente dans la mémoire RLDQ.

Et, ces premiers moyens de commande MCTL1 vont, en présence d'une telle instruction de chargement LDR<sub>x</sub> temporellement plus ancienne stockée dans la première mémoire RLDQ, extraire l'instruction opérative DUOp<sub>x</sub> dès que l'instruction de chargement LDR<sub>x</sub> a été extraite de la mémoire RLDQ.

Dans un mode de réalisation, tel que celui illustré sur la figure 3, les premiers moyens de commande MCTL1 comportent un premier compteur de lecture RDCTR1, circulaire, incrémenté à chaque fois qu'une instruction, quelle qu'elle soit, est extraite de la deuxième mémoire DIDQ.

Les moyens MCTL1 comportent par ailleurs un premier compteur d'écriture WRCTR1, circulaire, incrémenté à chaque fois qu'une instruction, quelle qu'elle soit, est stockée dans la deuxième mémoire DIDQ.

Dans l'exemple décrit ici, les deux compteurs ont une taille identique. Cette taille identique est un nombre de bits nécessaire à coder en binaire la profondeur de la deuxième mémoire DIDQ. Dans l'exemple illustré, puisque la mémoire DIDQ a une profondeur de quatre, les compteurs sont des compteurs sur deux bits, c'est-à-dire que la valeur courante WRVL1 du premier compteur d'écriture WRCTR1 et la valeur courante RDVL1 du premier compteur de lecture RDCTR1 sont des mots



de deux bits. Bien entendu la profondeur de la FIFO peut être plus importante. Par ailleurs, à chaque premier compteur est associé un bit de débordement (WROVFL1 pour le premier compteur d'écriture et RDOVFL1 pour le compteur de lecture). Ce bit de débordement change de valeur à chaque fois que le premier compteur correspondant revient à sa valeur initiale. En d'autres termes, au rythme des fenêtres circulaires successives de comptage, le bit de débordement prend successivement les valeurs 0 et 1.

40 Outre le premier compteur d'écriture et le premier compteur de lecture, les premiers moyens de commande MCTL1 comportent un ensemble de premiers registres élémentaires RT0-RT15, respectivement associés à l'ensemble des registres R0-R15. Ainsi, le registre élémentaire RTx est associé au registre Rx portant le numéro x.

15 Chaque registre élémentaire RTx comporte un champ principal LCHx, ainsi qu'un premier champ auxiliaire LBAX et un deuxième champ auxiliaire LBBx. On reviendra plus en détail sur la signification et la fonction de ces champs auxiliaires qui, dans l'exemple décrit ici, sont des champs à un bit.

20 Les premiers moyens de commande MCTL1 comportent, par ailleurs, quatre blocs de commande MC00-MC13 dont on va maintenant décrire plus en détail la fonctionnalité en se référant aux figures 4 et suivantes.

Le premier bloc de commande MC00 est apte, à chaque fois qu'une instruction de chargement LDRx concernant le registre Rx, est stockée dans la première mémoire RLDQ (étape 400, figure 4), à stocker la valeur courante WRVL1 du premier compteur d'écriture WRCTR1 dans le champ principal LCHx du premier registre élémentaire RTx associé au registre Rx. Par ailleurs, le bit LBBx (deuxième champ auxiliaire) du premier registre élémentaire RTx, initialement à 0, prend la valeur 1 pour indiquer qu'une instruction de chargement LDRx a été effectivement stockée dans la mémoire RLDQ (étape 410).

Enfin, la valeur courante du bit de débordement WROVFL1 du premier compteur d'écriture est stockée dans le premier champ auxiliaire (un bit) LBAX du premier registre élémentaire RTx.

En fait, la valeur stockée dans le champ principal LCHx du premier registre élémentaire RTx, dont la taille en bit est égale à la profondeur de la première mémoire (ici deux bits), représente la position "symbolique" de la dernière instruction de chargement LDRx écrite dans la mémoire RLDQ. Le premier champ auxiliaire LBAX, destiné à recevoir la valeur courante du bit de dépassement du premier compteur d'écriture, permet de prendre en compte et de gérer les positions éventuelles respectives des pointeurs dans des fenêtres circulaires de comptage différentes.

10 Dans l'étape 420, on va maintenant déterminer si l'instruction de chargement LDRx est toujours présente dans la mémoire RLDQ, ou bien si elle a quitté cette mémoire. Ceci est effectué par le deuxième bloc de commande MC11 et va maintenant être décrit plus en détail en se référant plus particulièrement à la figure 5.

15 Basiquement, le deuxième bloc de commande MC11 comporte des premiers moyens de comparaison MC100 qui vont comparer le contenu du champ principal LCHx du premier registre élémentaire RTx avec la valeur courante RDVL1 du premier compteur de lecture. Cette comparaison va permettre de déterminer si l'instruction LDRx est toujours présente ou a quitté la mémoire RLDQ.

Ceci étant, afin de prendre en compte les effets des fenêtres circulaires de comptage, il est prévu que le deuxième bloc de commande MC11 comporte également des premiers moyens de comparaison auxiliaires, formés ici d'une première porte logique NON OU EXCLUSIF (XNOR), référencée PLO10. Cette porte logique est destinée à élaborer un premier signal logique SW10 à partir du contenu du premier champ auxiliaire LBAX et de la valeur du bit de dépassement RDOVFL1 du premier compteur de lecture.

En d'autres termes, ce signal logique SW10 indique si le pointeur de lecture PL2 est dans la même fenêtre circulaire que la position de la dernière instruction LDRx affectant le registre Rx.

On teste alors, dans l'étape 4200, la valeur du premier signal logique SW10. Si ce signal logique est à 0, on teste, dans l'étape 4220, si la valeur du mot contenu dans le champ principal LCHx est strictement supérieure à la valeur courante RDVL1 du premier compteur de lecture. Si



Si tel est le cas, alors cela signifie que l'instruction LDRx a quitté la mémoire RLDQ (étape 4230) et on met alors le bit LBBx à 0 (étape 430, figure 4).

Si, par contre, la valeur du mot contenu dans le champ principal LCHx n'est pas strictement supérieure à la valeur courante RDVL1 du premier compteur de lecture, alors cela signifie que l'instruction de chargement LDRx est toujours présente dans la mémoire RLDQ (étape 4240). Dans ce cas, le bit LBBx est maintenu à 1.

Si, par contre, dans l'étape 4200, le signal logique SW10 est à 1, les premiers moyens de comparaison MC100 vont tester si la valeur courante du premier compteur de lecture RDVL1 est strictement supérieure au mot contenu dans le champ principal LCHx du premier registre élémentaire RTx (étape 4210).

Si tel est le cas, cela signifie que l'instruction de chargement LDRx a quitté la mémoire RLDQ (étape 4230). Dans ce cas, le bit LBBx est mis à 0 (étape 430, figure 4).

Dans le cas contraire, cela signifie (étape 4240) que l'instruction de chargement LDRx est toujours présente dans la mémoire RLDQ.

On va maintenant décrire plus en détail, en se référant plus particulièrement aux figures 6 et 7, les traitements effectués lorsqu'une instruction opérative DUOPx est stockée dans le champ utile LCHU1 de l'étage d'entrée LETG1 de la mémoire DIDQ, ainsi que l'élaboration de son critère d'extraction, lorsque cette instruction opérative DUOPx a atteint la tête de la mémoire DIDQ, c'est-à-dire lorsqu'elle est présente dans le champ utile LCHU4 de l'étage de sortie LETG4.

A des fins de simplification, on suppose dans un premier temps que l'instruction opérative DUOPx n'implique qu'un seul registre, à savoir le registre Rx.

Afin de ne pas perturber le bon synchronisme temporel des traitements, à chaque fois qu'une instruction opérative DUOPx est stockée dans la mémoire DIDQ, et si simultanément une instruction de chargement n'est pas stockée dans la mémoire RLDQ, on stocke également dans la mémoire RLDQ une instruction non-opérative NOP. Comme cela est connu de l'homme du métier, une instruction NOP ne modifie pas le contenu des registres architecturaux du coeur de processeur.

Par ailleurs, à chaque fois qu'une instruction opérative DUOPx



est stockée dans la mémoire DIDQ, par exemple dans le champ utile LCHU1 de l'étage LETG1, on lui associe une étiquette qui correspond au contenu du registre élémentaire RTx correspondant.

Et, cette étiquette va continuellement être attachée à l'instruction opérative DUOPx dans la mémoire DIDQ.

En pratique (figure 3), l'étiquette associée à une instruction opérative DUOPx est formée dans chaque étage de la mémoire DIDQ par

- un premier champ supplémentaire LDtag,

- un deuxième champ supplémentaire (1 bit) AuxLD, et

- un troisième champ supplémentaire (1 bit également) ActLD.

Lorsqu'une instruction opérative DUOPx est stockée dans l'étage de tête de la mémoire DIDQ (étape 600, figure 6), le troisième bloc de commande MC12 transfère (étape 610, figure 6)

le contenu du champ principal LCHx dans le premier champ supplémentaire LDtag,

le contenu du premier champ auxiliaire LBx dans le deuxième champ supplémentaire AuxLD, et,

le contenu du deuxième champ auxiliaire LBBx dans le troisième champ supplémentaire ActLD.

Et, le contenu de cette étiquette va rester inchangé lors de la propagation de l'instruction DUOPx à laquelle elle est rattachée, jusqu'à ce que cette instruction atteigne la tête de la mémoire DIDQ.

A ce moment-là, le quatrième bloc de commande MC13 va tester (étape 620) la présence ou l'absence d'une instruction de chargement LDRx temporellement plus ancienne dans la mémoire RLDQ.

Si une instruction de chargement LDRx est présente, alors le troisième bloc de commande MC12 va attendre, pour extraire l'instruction opérative de la mémoire DIDQ, que l'instruction de chargement LDRx ait été extraite de la mémoire RLDQ (étape 640).

Si, par contre, aucune instruction LDRx n'est détectée dans la mémoire RLDQ, le troisième bloc de commande MC12 va extraire immédiatement l'instruction opérative DUOPx de la tête de la mémoire DIDQ (étape 630).

Et, c'est le quatrième bloc de commande MC13 qui va élaborer le critère d'extraction de cette instruction opérative DUOPx ayant atteint la

tête de la deuxième mémoire.

Pour déterminer si une instruction de chargement LDRx est présente dans la mémoire RLDQ, le bloc MC13 va tout d'abord tester la valeur du bit ActLD de l'étiquette attachée à l'instruction opérative DUOPx (étape 6200, figure 7).

Si ce bit ActLD est égal à 0, alors cela signifie qu'aucune instruction LDRx temporellement plus ancienne que l'instruction DUOPx n'est présente dans la mémoire RLDQ (étape 6210). Dans ce cas, l'instruction opérative DUOPx peut être immédiatement extraite de la mémoire DIDQ (étape 630, figure 6).

Si, par contre, le bit ActLD est à 1, alors il convient basiquement de comparer le contenu du premier champ supplémentaire LDtag avec la valeur courante RDVLI du premier compteur de lecture. C'est le rôle des deuxièmes moyens de comparaison MC300.

Et, basiquement, une absence d'instruction LDRx se traduira par une valeur RDVLI strictement supérieure au premier champ supplémentaire LDtag.

Ceci étant, là encore, il convient de prendre en compte les effets des fenêtres circulaires de comptage.

C'est la raison pour laquelle il est prévu que le bloc de commande MC13 comporte des premiers moyens de comparaison supplémentaires, formés ici d'une deuxième porte logique NON OU EXCLUSIF, référencée PLO30, et apte à comparer le bit AuxLD avec le bit de dépassement RDOVLI du premier compteur de lecture. Le résultat de cette comparaison fournit un deuxième signal logique SW30. Ce signal SW30 indique si le pointeur de lecture PL1 de la première mémoire RLDQ est dans la même fenêtre circulaire que la position de la dernière instruction de chargement LDRx, ayant potentiellement un effet sur l'instruction opérative DUOPx.

Si le signal SW30 est à 1, alors l'instruction opérative DUOPx pourra être transmise si la valeur RDVLI est supérieure à LDtag (étape 6230, figure 7).

Dans le cas contraire, cela signifie qu'une instruction LDRx est toujours présente dans la première mémoire RLDQ (étape 6250). Et, il convient d'attendre avant d'extraire l'instruction opérative DUOPx de la

tête de la mémoire DIDQ.

Si le signal SW30 est à 0, alors on pourra extraire l'instruction opérative DUOPx si la valeur du mot LDtag est strictement supérieure à la valeur du premier compteur de lecture RDVL1 (étape 6240).

5 Dans le cas contraire, il conviendra d'attendre (étape 6250).

Matériellement, les différents blocs de commande MC10-MC13 peuvent être réalisés sous la forme d'un circuit intégré en utilisant des outils de synthèse logique.

Dans le cas où une instruction opérative DUOPxy implique plus  
10 d'un registre, par exemple 2 registres Rx et Ry, le troisième bloc de commande M12 va, lorsque ladite instruction opérative est stockée dans la deuxième mémoire DIDQ, sélectionner parmi tous les premiers registres élémentaires RTx, RTy associés à tous les registres Rx, Ry impliqués dans  
15 ladite instruction opérative DUOPxy, celui correspondant à la dernière (c'est-à-dire temporellement la plus jeune) instruction de chargement modifiant l'un au moins de ces registres impliqués.

Et, c'est alors le contenu du champ principal du premier registre  
élémentaire sélectionné, par exemple RTx, qui va être transféré dans le  
premier champ supplémentaire LDtag de l'étage d'entrée de la mémoire  
20 DIDQ.

Matériellement, les différents blocs de commande MC00-MC13 peuvent être réalisés sous la forme d'un circuit intégré en utilisant des outils de synthèse logique.

Le mode de réalisation qui vient d'être décrit permet une mise en  
25 oeuvre très simple du procédé selon l'invention. Il convient de noter également que la profondeur de la mémoire RLDQ peut être différente de celle de la mémoire DIDQ.

Par ailleurs tout ce qui vient d'être décrit pour un registre Rx, est effectué en réalité en parallèle simultanément pour tous les registres.

30 Dans le mode de réalisation illustré sur la figure 8, on suppose maintenant que le fonctionnement du processeur incorpore le principe des instructions gardées. Les instructions gardées peuvent être destinées à des unités de traitement différentes de l'unité DU, par exemple l'unité AU ou l'unité GU. Elles peuvent être également destinées à l'unité de traitement  
35 DU. Et, parmi les instructions gardées destinées à l'unité de traitement

10 - Dans l'unité DU, on trouve des instructions gardées de chargement dans un registre de  
 15 - données mémorisées. Le moyen de mémoire de l'unité DU comporte en outre, comme  
 20 - on le verra plus en détail ci-après, une troisième mémoire SNDGQ  
 25 - destinée à stocker des instructions de transmission et une quatrième  
 30 - mémoire RCGQ destinée à stocker toutes les valeurs d'indications de  
 35 - garde associées aux instructions de chargement gardées, et issues du  
 40 - registre des indications de garde GR. L'unité DU comporte le registre des indications de garde GR. Ce  
 45 - registre stocke les valeurs courantes de toutes les indications de garde, qui  
 50 - sont par exemple au nombre de seize. On rappelle ici qu'une instruction gardée, signifie que cette  
 55 - instruction est affectée d'une indication de garde, prise parmi les seize  
 60 - indications de garde du registre GR, et, en fonction de la valeur de cette  
 65 - indication de garde, l'instruction sera ou non exécutée. On se réfère maintenant plus particulièrement à la figure 9, pour  
 70 - décrire un mécanisme particulièrement avantageux de gestion de ces  
 75 - instructions gardées, selon l'invention. Lorsqu'une instruction gardée est délivrée au moyen de mémoire  
 80 - AIDQ d'une unité de traitement, par exemple l'unité AU ou l'unité GU, qui  
 85 - ne détient pas le registre d'indications de garde GR, cette délivrance  
 90 - provoque la délivrance au moyen de mémoire de l'unité de traitement DU,  
 95 - qui contient le registre des indications de garde, d'une instruction dite "de  
 100 - transmission" destinée à faire transmettre à l'unité AU ou GU la valeur de  
 105 - l'indication de garde associée à ladite instruction gardée qui a été délivrée  
 110 - à l'unité AU ou GU. Par ailleurs, lorsqu'une instruction de chargement gardée est  
 115 - délivrée à l'unité DU, une instruction de transmission SNDGx destinée à  
 120 - faire transmettre à la quatrième mémoire REGQ la valeur de l'indication  
 125 - de garde associée à ladite instruction gardée, est également stockée dans  
 130 - la mémoire SNDGQ. Par contre, toute autre instruction gardée, par exemple une  
 135 - instruction opérative gardée, délivrée à l'unité DU, est stockée dans la  
 140 - mémoire DIDQ et ne donne pas lieu à la délivrance d'une instruction de  
 145 - transmission SNDGx.

Sur la figure 9, la référence SNDGx désigne une telle instruction de transmission, et x désigne le numéro de l'indication de garde Gx associée à l'instruction gardée qui a été délivrée à l'une des unités AU ou GU, ou bien associée à une instruction de chargement gardée qui peut être celle référencée LDRx (impliquant le registre Rx), ou bien référencée LDRy (impliquant un registre Ry). Dans la suite du texte, à des fins de simplification, on supposera que l'indication Gx est associée le cas échéant à l'instruction LDRx.

La troisième mémoire SNDGQ comporte plusieurs étages (ici quatre), référencés ETG1-ETG4. Chaque étage ETGi comporte un champ utile CHUi pour le stockage d'une instruction de transmission SNDGx. L'écriture et la lecture dans la troisième mémoire SNDGQ sont gérées par un pointeur d'écriture PE3 et un pointeur de lecture PL3. Une instruction de transmission SNDGx qui aurait été stockée dans l'étage ETG1, va alors se propager séquentiellement depuis l'étage d'entrée ETG1 de la mémoire jusqu'à l'étage de sortie ETG4, ou étage de tête, de la mémoire SNDGQ. Bien entendu si, à un instant donné, la FIFO est partiellement remplie ou vide, une instruction pourra être stockée directement au premier étage disponible, par exemple ETG2, ou ETG3 ou bien ETG4, comme cela est bien connu de l'homme du métier. Ainsi au sens de la présente invention l'étage d'entrée n'est donc pas nécessairement l'étage ETG1.

La deuxième mémoire DIDQ est destinée à stocker toutes les autres instructions INST destinées à l'unité DU, à l'exception des instructions de chargement LDRx, et notamment les instructions dites "modificatrices", c'est-à-dire destinées à modifier la valeur d'une indication de garde stockée dans le registre des indications de garde GR. Une instruction modificatrice destinée à modifier la valeur de l'indication de garde portant le numéro x, est référencée dans la suite du texte par GMix.

Les deux mémoires SNDGQ et DIDQ ont, dans l'exemple décrit ici, la même profondeur (taille) égale à quatre. Il convient également de noter que les pointeurs PE3 et PL3 associés à la mémoire SNDGQ sont décorrélés (c'est-à-dire commandés de façon indépendante) par rapport aux pointeurs PE2 et PL2 associés à la mémoire DIDQ.

MCTL2, par exemple incorporés au sein de l'unité DU. Ces deuxièmes moyens de contrôle vont permettre, comme on va le voir plus en détail ci-après, de délivrer les instructions de transmission SNDGx beaucoup plus rapidement aux étages d'exécution pipelinés de l'unité DU, et par conséquent au registre des indications de garde GR, de façon à permettre la délivrance de la valeur de l'indication de garde associée à l'instruction gardée qui est en attente d'exécution.

Enfin, ces deuxièmes moyens de commande MCTL2 vont permettre d'extraire, dans le moyen de mémoire associé à une autre unité de traitement, par exemple l'unité GU, ou bien

10 dans la mémoire RLDQ.

Plus précisément, ces deuxièmes moyens de commande MCTL2 vont extraire de la mémoire SNDGQ une instruction de transmission SNDGx ayant atteint la tête ETG4 de cette mémoire SNDGQ, si aucune instruction GMix temporellement plus ancienne destinée à modifier la

15 valeur de l'indication de garde Gx associée à cette instruction de transmission SNDGx, n'est présente dans la mémoire DIDQ.

Et, ces moyens de commande MCTL2 vont, en présence d'une telle instruction modificatrice GMix temporellement plus ancienne stockée dans la deuxième mémoire DIDQ, extraire l'instruction de

20 transmission SNDGx dès que l'instruction modificatrice GMix a été extraite de la mémoire DIDQ.

Dans un mode de réalisation, tel que celui illustré sur la figure 9, les deuxièmes moyens de commande MCTL2 comportent un deuxième compteur de lecture RDCTR2, circulaire, incrémenté à chaque fois qu'une instruction, quelle qu'elle soit, est extraite de la deuxième mémoire DIDQ.

Les moyens MCTL2 comportent par ailleurs un deuxième compteur d'écriture WRCTR2, circulaire, incrémenté à chaque fois qu'une instruction, quelle qu'elle soit, est stockée dans la deuxième mémoire DIDQ.

Dans l'exemple décrit ici, les deux compteurs ont une taille identique. Cette taille identique est un nombre de bits nécessaire à coder en binaire la profondeur de la deuxième mémoire DIDQ. Dans l'exemple illustré, puisque la mémoire DIDQ a une profondeur de quatre, les deuxièmes compteurs sont des compteurs sur deux bits, c'est-à-dire que la

la valeur courante WRVL2 du deuxième compteur d'écriture WRCTR2 et la valeur courante RDVL2 du deuxième compteur de lecture RDCTR2 sont stockées dans des mots de deux bits. Bien entendu la profondeur de la FIFO peut être plus importante.

5 Par ailleurs, à chaque deuxième compteur est associé un bit de dépassement (WROVFL2 pour le compteur d'écriture et RD OVFL2 pour le compteur de lecture). Ce bit de dépassement change de valeur à chaque fois que le compteur correspondant revient à sa valeur initiale. En d'autres termes, au rythme des fenêtres circulaires successives de comptage, le bit de dépassement prend successivement les valeurs 0 et 1.

10 Outre le compteur d'écriture et le compteur de lecture, les moyens de commande MCTL2 comportent un ensemble de registres élémentaires GT0-GT15, respectivement associés à l'ensemble des indications de garde G0-G15. Ainsi, le registre élémentaire GTx est associé à l'indication de garde Gx portant le numéro x.

15 Chaque registre élémentaire GTx comporte un champ principal CHx, ainsi qu'un premier champ auxiliaire BAx et un deuxième champ auxiliaire BBx. On reviendra plus en détail sur la signification et la fonction de ces champs auxiliaires qui, dans l'exemple décrit ici, sont des champs à un bit.

Les moyens de commande MCTL2 comportent, par ailleurs, quatre blocs de commande MC0-MC3 dont on va maintenant décrire plus en détail la fonctionnalité en se référant aux figures 10 et suivantes.

20 Le cinquième bloc de commande MC0 est apte, à chaque fois qu'une instruction GMix modifiant la valeur de l'indication de garde Gx, est stockée dans la deuxième mémoire DIDQ (étape 40, figure 10), à stocker la valeur courante WRVL2 du compteur d'écriture WRCTR2 dans le champ principal CHx du registre élémentaire GTx associé à l'indication de garde Gx. Par ailleurs, le bit BBx (deuxième champ auxiliaire) du registre élémentaire GTx, initialement à 0, prend la valeur 1 pour indiquer qu'une instruction modificatrice GMix a été effectivement stockée dans la 25 mémoire DIDQ (étape 41).

30 Enfin, la valeur courante du bit de dépassement WROVFL2 du compteur d'écriture est stockée dans le premier champ auxiliaire (un bit) BAx du registre élémentaire GTx.

En fait, la valeur stockée dans le champ principal CHx du registre élémentaire GTx, dont la taille en bit est égale à la profondeur de la deuxième mémoire (ici deux bits), représente la position "symbolique" de la dernière instruction modificatrice GMix écrite dans la mémoire DIDQ. Le premier champ auxiliaire BAx, destiné à recevoir la valeur courante du bit de dépassement du compteur d'écriture, permet de prendre en compte et de gérer les positions éventuelles respectives des pointeurs dans des fenêtres circulaires de comptage différentes.

Dans l'étape 42, on va maintenant déterminer si l'instruction modificatrice GMix est toujours présente dans la mémoire DIDQ, ou bien si elle a quitté cette mémoire. Ceci est effectué par le sixième bloc de commande MC1 et va maintenant être décrit plus en détail en se référant plus particulièrement à la figure 11.

Basiquement, le sixième bloc de commande MC1 comporte des deuxièmes moyens de comparaison MC10 qui vont comparer le contenu du champ principal CHx du registre élémentaire GTx avec la valeur courante RDVL2 du compteur de lecture. Cette comparaison va permettre de déterminer si l'instruction GMix est toujours présente ou a quitté la mémoire DIDQ.

Ceci étant, afin de prendre en compte les effets des fenêtres circulaires de comptage, il est prévu que le sixième bloc de commande MC1 comporte également des deuxièmes moyens de comparaison auxiliaires, formés ici d'une première porte logique NON OU EXCLUSIF (XNOR), référencée PLO1. Cette porte logique est destinée à élaborer un premier signal logique SW1 à partir du contenu du premier champ auxiliaire BAx et de la valeur du bit de dépassement RDOVFL2 du compteur de lecture.

En d'autres termes, ce signal logique SW1 indique si le pointeur de lecture PL2 est dans la même fenêtre circulaire que la position de la dernière instruction GMix affectant l'indication de garde Gx.

On teste alors, dans l'étape 420, la valeur du premier signal logique SW1. Si ce signal logique est à 0, on teste, dans l'étape 422, si la valeur du mot contenu dans le champ principal CHx est strictement supérieure à la valeur courante RDVL2 du compteur de lecture. Si tel est le cas, alors cela signifie que l'instruction GMix a quitté la mémoire DIDQ



(étape 423) et on met alors le bit BBx à 0 (étape 43, figure 10).

Si, par contre, la valeur du mot contenu dans le champ principal CHx n'est pas strictement supérieure à la valeur courante RDVL2 du compteur de lecture, alors cela signifie que l'instruction modificatrice GMix est toujours présente dans la mémoire DIDQ (étape 424). Dans ce cas, le bit BBx est maintenu à 1.

Si, par contre, dans l'étape 420, le signal logique SW1 est à 1, les moyens de comparaison MC10 vont tester si la valeur courante du compteur de lecture RDVL2 est strictement supérieure au mot contenu dans le champ principal CHx du registre élémentaire GTx (étape 421).

Si tel est le cas, cela signifie que l'instruction modificatrice GMix a quitté la mémoire DIDQ (étape 423). Dans ce cas, le bit BBx est mis à 0 (étape 43, figure 10).

Dans le cas contraire, cela signifie (étape 424) que l'instruction modificatrice GMix est toujours présente dans la mémoire DIDQ.

On va maintenant décrire plus en détail, en se référant plus particulièrement aux figures 12 et 13, les traitements effectués lorsqu'une instruction de transmission SNDGx est stockée dans le champ utile CHU1 de l'étage d'entrée ETG1 de la mémoire SNDGQ, ainsi que l'élaboration de son critère d'extraction, lorsque cette instruction de transmission SNDGx a atteint la tête de la mémoire SNDGQ, c'est-à-dire lorsqu'elle est présente dans le champ utile CHU4 de l'étage de sortie ETG4.

Afin de ne pas perturber le bon synchronisme temporel des traitements, à chaque fois qu'une instruction de transmission SNDGx est stockée dans la mémoire SNDGQ et que l'on ne stocke pas simultanément une autre instruction dans la mémoire DIDQ, on stocke également dans la mémoire DIDQ une instruction non-opérative NOP.

Par ailleurs, à chaque fois qu'une instruction de transmission SNDGx est stockée dans la mémoire SNDGQ, par exemple dans le champ utile CHU1 de l'étage ETG1, on lui associe une étiquette qui correspond au contenu du registre élémentaire GTx correspondant.

Et, cette étiquette va continuellement être attachée à l'instruction de transmission SNDGx dans la mémoire SNDGQ.

En pratique (figure 9), l'étiquette associée à une instruction de transmission SNDGx est formée dans chaque étage de la mémoire SNDGQ

par

- un premier champ supplémentaire Sndgtag,
- un deuxième champ supplémentaire (1 bit) Auxgmi, et
- un troisième champ supplémentaire (1 bit également) Actgmi.

5 Lorsque une instruction de transmission SNDGx est stockée dans l'étage de tête de la mémoire SNDGQ (étape 60, figure 12), le septième bloc de commande MC2 transfère (étape 61, figure 12)

le contenu du champ principal CHx dans le premier champ supplémentaire Sndgtag,

10 - le contenu du premier champ auxiliaire BAx dans le deuxième champ supplémentaire Auxgmi, et,

- le contenu du deuxième champ auxiliaire BBx dans le troisième champ supplémentaire Actgmi.

15 Et, le contenu de cette étiquette va rester inchangé lors de la propagation de l'instruction SNDGx à laquelle elle est rattachée, jusqu'à ce que cette instruction atteigne la tête de la mémoire SNDGQ.

A ce moment-là, le huitième bloc de commande MC3 va tester (étape 62) la présence ou l'absence d'une instruction modificatrice GMix temporellement plus ancienne dans la mémoire DIDQ.

20 Si une instruction modificatrice GMix est présente, alors le troisième bloc de commande MC2 va attendre, pour extraire l'instruction de transmission de la mémoire SNDGQ, que l'instruction modificatrice GMix ait été extraite de la mémoire DIDQ (étape 64).

25 Si, par contre, aucune instruction GMix n'est détectée dans la mémoire DIDQ, le septième bloc de commande MC2 va extraire immédiatement l'instruction de transmission SNDGx de la tête de la mémoire SNDQ (étape 63).

30 Et, c'est le huitième bloc de commande MC3 qui va élaborer le critère d'extraction de cette instruction de transmission SNDGx ayant atteint la tête de la première mémoire.

Pour déterminer si une instruction modificatrice GMix est présente dans la mémoire DIDQ, le bloc MC3 va tout d'abord tester la valeur du bit Actgmi de l'étiquette attachée à l'instruction de transmission SNDGx (étape 620, figure 13).

35 Si ce bit Actgmi est égal à 0, alors cela signifie qu'aucune instruction GMix temporellement plus ancienne que l'instruction SNDGx

n'est présente dans la mémoire DIDQ (étape 621). Dans ce cas, l'instruction de transmission SNDGx peut être immédiatement extraite de la mémoire SNDGQ (étape 63, figure 12).

Si, par contre, le bit Actgmi est à 1, alors il convient basiquement de comparer le contenu du premier champ supplémentaire Sndgtag avec la valeur courante RDVL2 du compteur de lecture. C'est le rôle des moyens de comparaison MC30.

Et, basiquement, une absence d'instruction GMix se traduira par une valeur RDVL2 strictement supérieure au premier champ supplémentaire Sndgtag.

Ceci étant, là encore, il convient de prendre en compte les effets des fenêtres circulaires de comptage.

C'est la raison pour laquelle il est prévu que le bloc de commande MC3 comporte des deuxièmes moyens de comparaison supplémentaires, formés ici d'une deuxième porte logique NON OU EXCLUSIF, référencée PLO3, et apte à comparer le bit Auxgmi avec le bit de dépassement RDOVFL2 du compteur de lecture. Le résultat de cette comparaison fournit un deuxième signal logique SW3. Ce signal SW3 indique si le pointeur de lecture PL2 de la deuxième mémoire DIDQ est dans la même fenêtre circulaire que la position de la dernière instruction modificatrice GMix, ayant potentiellement un effet sur l'instruction de transmission SNDGx.

Si le signal SW3 est à 1, alors l'instruction de transmission SNDGx pourra être transmise si la valeur RDVL est supérieure à Sndgtag (étape 623, figure 7).

Dans le cas contraire, cela signifie qu'une instruction GMix est toujours présente dans la deuxième mémoire DIDQ (étape 625). Et, il convient d'attendre avant d'extraire l'instruction de transmission SNDGx de la tête de la mémoire SNDGQ.

Si le signal SW3 est à 0, alors on pourra extraire l'instruction de transmission SNDGx si la valeur du mot Sndgtag est strictement supérieure à la valeur du compteur de lecture RDVL (étape 624).

Dans le cas contraire, il conviendra d'attendre (étape 625).

Matériellement, les différents blocs de commande MC0-MC3 peuvent être réalisés, sous la forme d'un circuit intégré en utilisant des

des outils de synthèse logique, selon les besoins.

Il convient de noter également que la profondeur de la mémoire SNDGQ peut être différente de celle de la mémoire DIDQ.

Par ailleurs, tout ce qui vient d'être décrit pour une indication de garde Gx, est effectué en réalité en parallèle simultanément pour toutes les indications de garde.

Sur la figure 14, qui illustre à nouveau les quatre mémoires FIFO selon l'invention, permettant la gestion globale d'instructions selon l'invention, y compris les instructions de chargement de données mémorisées dans un registre, les instructions gardées, et en particulier les instructions de chargement gardées, on voit qu'une instruction de transmission destinée à faire transmettre la valeur de l'indication de garde associée à l'instruction de chargement qui a été stockée dans la mémoire RLDQ, va être délivrée au registre des indications de garde GR. Puis, la valeur correspondante de l'indication de garde va être stockée dans la quatrième mémoire RCGQ.

Bien entendu, comme déjà indiqué ci-avant, dans le cas d'une indication gardée qui a été délivrée à une autre unité de traitement, par exemple l'unité AU ou l'unité GU, la valeur de l'indication de garde associée à ces instructions gardées va être délivrée à cette autre unité AU ou GU.

Par ailleurs, sur la figure 14, les chiffres 1, 2 et 3 situés dans un cercle reprennent les dépendances "temporelles" des trois mémoires RLDQ, DIDQ et SNDGQ.

En d'autres termes, en présence d'une instruction modificatrice de chargement temporellement plus ancienne destinée à modifier la valeur d'au moins un registre associé à une instruction opérative, on extrait ladite instruction opérative de la deuxième mémoire DIDQ seulement après que l'instruction modificatrice de chargement a été extraite de la première mémoire RLDQ.

De même, en présence d'une instruction modificatrice temporellement plus ancienne destinée à modifier la valeur de l'indication de garde associée à une instruction de transmission, on extrait cette instruction de transmission de la troisième mémoire seulement après que l'instruction modificatrice a été extraite de la deuxième mémoire.

Par ailleurs, il est bien entendu que certaines instructions de chargement peuvent être non gardées. Dans ce cas, elles sont stockées dans la mémoire RLDQ et il n'y a pas d'instructions correspondantes de transmission stockées dans la mémoire SNDGQ. De plus, dans ce cas, on ne lit pas le contenu de l'étage de sortie de la quatrième mémoire RCGQ.

## REVENDICATIONS

1. Procédé de gestion d'instructions au sein d'un processeur à architecture découplée, le processeur comportant un coeur contenant plusieurs unités de traitement (DU, AU, GU) respectivement associées à des moyens de mémoire du type FIFO pour stocker séquentiellement les instructions respectives qui sont destinées aux unités correspondantes, procédé dans lequel la délivrance au coeur de processeur, d'une instruction de chargement d'une donnée mémorisée dans un registre, provoque la délivrance au moyen de mémoire d'une première unité de traitement (DU) d'une instruction de chargement dans ledit registre et la délivrance au moyen de mémoire d'une deuxième unité de traitement (AU) d'une instruction destinée à élaborer ladite donnée mémorisée, l'instruction de chargement n'étant exécutée que lorsque ladite donnée mémorisée a été délivrée par ladite deuxième unité de traitement, caractérisé par le fait que le moyen de mémoire de la première unité de traitement comporte une première mémoire de type FIFO (RLDQ), et une deuxième mémoire de type FIFO (DIDQ) distincte de la première, par le fait qu'on stocke chaque instruction de chargement (LDRx) dans la première mémoire (RLDQ) et certaines au moins des autres instructions opératives (INST) destinées à la première unité (DU) dans la deuxième mémoire (DIDQ), par le fait qu'on extrait de la deuxième mémoire (DIDQ) une instruction opérative impliquant au moins un registre (DUOPx) et ayant atteint la tête (ETG4) de cette deuxième mémoire, si aucune instruction de chargement (LDRx) temporellement plus ancienne destinée à modifier la valeur du ou des registres associés à cette instruction opérative, n'est présente dans la première mémoire, et par le fait qu'en présence d'une telle instruction modificatrice de chargement (LDRx) temporellement plus ancienne, on extrait ladite instruction opérative (DUOPx) de la deuxième mémoire (DIDQ) seulement après que l'instruction modificatrice de chargement (LDRx) a été extraite de la première mémoire (RLDQ).
2. Procédé selon la revendication 1, caractérisé par le fait qu'à chaque fois qu'on stocke dans la deuxième mémoire (DIDQ) une instruction sans qu'il soit prévu de stocker simultanément dans la première mémoire (RLDQ) une instruction de chargement (LDRx), on stocke dans la

première mémoire (RLDQ) une instruction non-opérative (NOP), par le fait qu'à chaque fois qu'on extrait une instruction (LDRx, NOP) de la première mémoire (RLDQ), on incrémente un premier compteur de lecture (RDCTR1), par le fait qu'à chaque fois qu'on stocke une instruction (INST) dans la première mémoire (RLDQ), on incrémente un premier compteur d'écriture (WRCTR1), par le fait qu'à chaque fois qu'on stocke dans la première mémoire (RLDQ) une instruction de chargement (LDRx), on mémorise la valeur courante (WRVL1) du premier compteur d'écriture, et par le fait que la détermination du caractère toujours présent de cette instruction de chargement (LDRx) dans la première mémoire prend en compte le résultat de la comparaison de ladite valeur courante mémorisée du premier compteur d'écriture avec la valeur courante du premier compteur de lecture (RDVL1).

3. Procédé selon la revendication 2, caractérisé par le fait que le premier compteur d'écriture (WRCTR1) et le premier compteur de lecture (RDCTR1) ont une taille binaire identique égale à la profondeur de la première mémoire (RLDQ), par le fait qu'on associe à chaque premier compteur un bit de dépassement (WROVFL1, RDOVFL1) changeant de valeur à chaque fois que le premier compteur correspondant revient à sa valeur initiale, par le fait qu'à chaque fois qu'on stocke dans la première mémoire une instruction de chargement (LDRx), on mémorise également la valeur courante du bit de dépassement (WROVFL1) du premier compteur d'écriture, et par le fait que la détermination du caractère toujours présent de cette instruction de chargement (LDRx) dans la première mémoire prend également en compte le résultat de la comparaison de la valeur courante du bit de dépassement (RDOVFL) du premier compteur de lecture avec ladite valeur mémorisée du bit de dépassement du premier compteur d'écriture.

4. Procédé selon la revendication 2 ou 3, caractérisé par le fait qu'à chaque fois qu'on stocke dans la deuxième mémoire (DIDQ) une instruction opérative (DUOPx) impliquant au moins un registre (DUOPx), on lui associe une première étiquette contenant la valeur courante du premier compteur d'écriture qui a été mémorisée lorsque la dernière instruction de chargement (LDRx) modifiant l'un au moins des registres impliqués dans ladite instruction opérative a été stockée dans la première mémoire, et par le fait que le critère d'extraction de cette instruction opérative

opérative (DUOPx) ayant atteint la tête (ETG4) de la deuxième mémoire (DIDQ) prend en compte le résultat de la comparaison entre ladite valeur courante mémorisée associée à cette instruction opérative, et la valeur courante du premier compteur de lecture (RDVL1):

5 Procédé selon les revendications 3 et 4, caractérisé par le fait qu'à chaque fois qu'on stocke une instruction opérative dans la deuxième mémoire, on lui associe également dans sa première étiquette la valeur courante du bit de dépassement du premier compteur d'écriture qui a été mémorisée lorsque la dernière instruction de chargement (LRDx) modifiant l'un au moins des registres impliqués dans ladite instruction opérative a été stockée dans la première mémoire, et par le fait que le critère d'extraction de cette instruction opérative ayant atteint la tête de la deuxième mémoire prend également en compte le résultat de la comparaison entre cette valeur courante mémorisée associée à cette instruction opérative et la valeur courante du bit de dépassement (RDOVFL1) du premier compteur de lecture.

6 Procédé selon l'une des revendications précédentes, caractérisé par le fait que le moyen de mémoire de la première unité comporte une troisième mémoire de type FIFO (SNDGQ), distincte de la première et de la deuxième mémoires, et une quatrième mémoire de type FIFO (RCGQ), distincte des trois premières, par le fait que la première unité (DU) contient un registre d'indications de garde (RG), par le fait que la délivrance d'une instruction gardée au moyen de mémoire (AIDQ) d'une autre unité de traitement différente de la première (AU ou GU), ou la délivrance d'une instruction de chargement gardée à la première unité, provoque la délivrance à la première unité de traitement (DU) et son stockage dans la troisième mémoire (SNDGQ), d'une instruction de transmission (SNDGx) destinée à faire transmettre à ladite autre unité (AU ou GU) ou à ladite quatrième mémoire (RCGQ), respectivement, la valeur de l'indication de garde (Gx) associée à ladite instruction gardée ou à ladite instruction de chargement gardée, respectivement, par le fait qu'on extrait de la troisième mémoire (SNDGQ) une instruction de transmission (SNDGx) ayant atteint la tête (ETG4) de cette troisième mémoire si aucune instruction modificatrice (GMIx) temporellement plus ancienne destinée à modifier la valeur de l'indication de garde (Gx) associée à cette



15 instruction de transmission (SNDGx), n'est présente dans la deuxième mémoire (DIDQ), et par le fait qu'en présence d'une telle instruction modificatrice (GMIX) temporellement plus ancienne, on extrait l'instruction de transmission (SNDGx) de la troisième mémoire (SNDGQ) seulement après que l'instruction modificatrice (GMIX) a été extraite de la deuxième mémoire (DIDQ).

20 Procédé selon la revendication 6, caractérisé par le fait qu'à chaque fois qu'on extrait une instruction (INST) de la deuxième mémoire (DIDQ), on incrémente un deuxième compteur de lecture (RDCTR2), par le fait qu'à chaque fois qu'on stocke une instruction (INST) dans la deuxième mémoire (DIDQ), on incrémente un deuxième compteur d'écriture (WRCTR2), par le fait qu'à chaque fois qu'on stocke dans la deuxième mémoire (DIDQ) une instruction (GMIX) modifiant la valeur d'une indication de garde (Gx), on mémorise la valeur courante (WRVL2) du deuxième compteur d'écriture, et par le fait que la détermination du caractère toujours présent de cette instruction modificatrice (GMIX) dans la deuxième mémoire prend en compte la comparaison de ladite valeur courante mémorisée du deuxième compteur d'écriture avec la valeur courante du deuxième compteur de lecture (RDVL2).

25 Procédé selon la revendication 7, caractérisé par le fait que le deuxième compteur d'écriture (WRCTR2) et le deuxième compteur de lecture (RDCTR2) ont une taille binaire identique égale à la profondeur de la deuxième mémoire (DIDQ), par le fait qu'on associe à chaque deuxième compteur un bit de dépassement (WROVFL2, RDOVFL2) changeant de valeur à chaque fois que le deuxième compteur correspondant revient à sa valeur initiale, par le fait qu'à chaque fois qu'on stocke dans la deuxième mémoire une instruction (GMIX) modifiant la valeur d'une indication de garde, on mémorise également la valeur courante du bit de dépassement (WROVFL2) du deuxième compteur d'écriture, et par le fait que la détermination du caractère toujours présent de cette instruction modificatrice (GMIX) dans la deuxième mémoire prend également en compte la comparaison de la valeur courante du bit de dépassement (RDOVFL2) du deuxième compteur de lecture avec ladite valeur mémorisée du bit de dépassement du deuxième compteur d'écriture.

30 Procédé selon la revendication 7 ou 8, caractérisé par le fait

qu'à chaque fois qu'on stocke une instruction de transmission (SNDGx) dans la troisième mémoire (SNDGQ), on lui associe une étiquette contenant la valeur courante du deuxième compteur d'écriture qui a été mémorisée lorsque la dernière instruction (GMIx) modifiant l'indication de garde (Gx) associée à ladite instruction de transmission a été stockée dans la deuxième mémoire, et par le fait que le critère d'extraction de cette instruction de transmission (SNDGx) ayant atteint la tête (ETG4) de la troisième mémoire (SNDGQ) prend en compte le résultat de la comparaison entre ladite valeur courante mémorisée associée à cette instruction de transmission, et la valeur courante du deuxième compteur de lecture (RDVL2).

10. Procédé selon les revendications 8 et 9, caractérisé par le fait qu'à chaque fois qu'on stocke une instruction de transmission dans la troisième mémoire, on lui associe également dans son étiquette la valeur courante du bit de dépassement du deuxième compteur d'écriture qui a été mémorisée lorsque la dernière instruction (GMIx) modifiant l'indication de garde associée à ladite instruction de transmission a été stockée dans la deuxième mémoire, et par le fait que le critère d'extraction de cette instruction de transmission ayant atteint la tête de la troisième mémoire prend également en compte le résultat de la comparaison entre cette valeur courante mémorisée associée à cette instruction de transmission et la valeur courante du bit de dépassement (RDVFL2) du deuxième compteur de lecture.

11. Procédure à architecture découplée, comprenant un coeur contenant plusieurs unités de traitement (DU, AU, GU) respectivement associées à des moyens de mémoire du type FIFO pour stocker séquentiellement les instructions respectives qui sont destinées aux unités correspondantes, et une unité centrale (CU) apte en réponse à une instruction de chargement d'une donnée mémorisée dans un registre, à délivrer au moyen de mémoire d'une première unité de traitement (DU), une instruction de chargement dans ledit registre, et à délivrer au moyen de mémoire d'une deuxième unité de traitement (AU) une instruction destinée à élaborer ladite donnée mémorisée, l'instruction de chargement n'étant exécutée que lorsque ladite donnée mémorisée a été délivrée par ladite deuxième unité de traitement, caractérisé par le fait que le moyen de mémoire de la pre-

première unité comporte une première mémoire de type FIFO (RLDQ), et une  
 deuxième mémoire de type FIFO (DIDQ) distincte de la première, et par le  
 fait que le processeur comporte des moyens d'aiguillage (DISP) aptes à stocker chaque  
 instruction de chargement (LDRx) dans la première mémoire (RLDQ) et  
 certaines au moins des autres instructions opératives (INST) destinées à la  
 première unité (DU) dans la deuxième mémoire, et des premiers moyens de commande (MCTL1) aptes  
 à extraire de la deuxième mémoire (DIDQ) une instruction opérative impliquant au moins un registre (DUOPx) et ayant atteint la tête  
 (ETG4) de cette deuxième mémoire, si aucune instruction de chargement (LDRx) temporellement plus ancienne destinée à modifier la valeur du ou  
 des registres associés à cette instruction opérative, n'est présente dans la  
 première mémoire (DIDQ) et en présence d'une telle instruction modificatrice de chargement (LDRx) temporellement plus ancienne, à extraire ladite instruction opérative (DUOPx) de la deuxième mémoire (SNDGQ) seulement après que  
 l'instruction modificatrice de chargement (LDRx) a été extraite de la première mémoire (RLDQ).

12. Processeur selon la revendication 11, caractérisé par le fait que les premiers moyens de commande (MCTL1) comportent un premier compteur de lecture (RDCTR1) incrémenté à chaque fois qu'une instruction est extraite de la première mémoire (RLDQ), un premier compteur d'écriture (WRCTR1) incrémenté à chaque fois qu'une instruction est stockée dans la première mémoire (RLDQ), un ensemble de premiers registres élémentaires (RT0-RT15) respectivement associés à l'ensemble des registres (R0-R15) capables d'être chargés par des données mémorisées, un premier bloc de commande (MC00) apte, à chaque fois qu'une instruction de chargement (LDRx) dans un registre (Rx) est stockée dans la première mémoire (RLDQ), à stocker la valeur courante (WRVL1) du premier compteur d'écriture dans un champ principal (LCHx) du premier registre élémentaire (RTx) associé à ce registre (Rx), un deuxième bloc de commande (MC11) apte à déterminer le caractère toujours présent de cette instruction de chargement (LDRx)

10 dans la première mémoire (RLDQ), et comportant des premiers moyens de  
comparaison (MC100) aptes à comparer le contenu du champ  
principal (LCHx) du premier registre élémentaire (RTx), avec la valeur  
courante (RDVL1) du premier compteur de lecture.

15 13. Processeur selon la revendication 12, caractérisé par le fait  
que le premier compteur d'écriture et le premier compteur de lecture ont  
une taille binaire identique égale à la profondeur de la première mémoire  
(RLDQ), par le fait qu'à chaque premier compteur est associé un bit de  
dépassément (WROVFL1, RDOVFL1) changeant de valeur à chaque fois  
20 que le premier compteur correspondant revient à sa valeur initiale, par le  
fait que chaque premier registre élémentaire (RTx) comporte en outre un  
champ auxiliaire à un bit (LBAX), par le fait que le premier bloc de com-  
mande (MC00) est apte, à chaque fois qu'une instruction de chargement  
(LDRx) dans un registre (Rx) est stockée dans la première mémoire  
25 (RLDQ), à stocker également la valeur courante du bit de dépassement  
(WROVFL1) du premier compteur d'écriture dans le champ auxiliaire  
(LBAX) du premier registre élémentaire correspondant (RTx), et par le  
fait que le deuxième bloc de commande (MC11) comporte des premiers  
moyens de comparaison auxiliaires (PLO10) aptes à comparer la valeur  
30 courante du bit de dépassement (RDOVFL1) du premier compteur de lec-  
ture avec le contenu du champ auxiliaire (LBAX).

14. Processeur selon la revendication 13, caractérisé par le fait  
que les premiers moyens de comparaison auxiliaires comportent une porte  
logique NON OU EXCLUSIF (PLO10).

25 15. Processeur selon l'une des revendications 12 à 14, caracté-  
risé par le fait que chaque étage (LETGi) de la deuxième mémoire (DIDQ)  
comporte un champ utile (LCHUi) pour le stockage d'une instruction opé-  
rative (DUOPx) impliquant au moins un registre (Rx) ou bien pour le stoc-  
30 kage d'une instruction non-opérative (NOP); et un premier champ supplé-  
mentaire (LDtag), par le fait que les moyens de commande (MCTL1) com-  
portent

un troisième bloc de commande (MC12) apte, à chaque fois  
qu'une instruction opérative (DUOPx) impliquant au moins un regis-  
35 tre (Rx) est stockée dans le champ utile (LCHU1) de l'étage d'entrée de la  
deuxième mémoire, à sélectionner parmi tous les premiers registres élé-

mentaires associés, à tous les registres impliqués dans ladite instruction opérative, celui correspondant à la dernière instruction de chargement (LDRx) modifiant l'un au moins de ces registres impliqués, et à transférer le contenu du champ principal (LCHx) du premier registre élémentaire sélectionné (RTx), dans le premier champ supplémentaire (LDtag) de l'étage d'entrée de la deuxième mémoire (DIDQ), et

un quatrième bloc de commande (MC13), apte à élaborer le critère d'extraction de cette instruction opérative (DUOPx) ayant atteint l'étage de tête (LETG4) de la deuxième mémoire (DIDQ), et comportant des premiers moyens de comparaison (MC300) aptes à comparer le contenu dudit premier champ supplémentaire (LDtag) de l'étage de tête avec la valeur courante (RDVL1) du premier compteur de lecture.

16. Processeur selon la revendication 15 prise en combinaison avec la revendication 13 ou 14, caractérisé par le fait que chaque étage de la deuxième mémoire comporte en outre un deuxième champ supplémentaire (AuxLD), par le fait que le troisième bloc de commande (MC12) est apte, à chaque fois qu'une instruction opérative est stockée dans le champ utile (LCHU1) de l'étage d'entrée de la deuxième mémoire, à transférer le contenu du champ auxiliaire (LBAX) du premier registre élémentaire sélectionné (RTx), dans le deuxième champ supplémentaire (AuxLD) de l'étage d'entrée de la deuxième mémoire, et par le fait que le quatrième bloc de commande (MC13) comportent des premiers moyens de comparaison supplémentaires (PLO30) aptes à comparer le contenu du deuxième champ supplémentaire (AuxLD) avec la valeur courante du bit de dépassement (RDOVFL1) du premier compteur de lecture.

17. Processeur selon la revendication 16, caractérisé par le fait que les premiers moyens de comparaison supplémentaires comportent une porte logique NON OU EXCLUSIF (PLO30).

18. Processeur selon l'une des revendications 11 à 17, caractérisé par le fait que le moyen de mémoire de la première unité (DU) comporte une troisième mémoire de type FIFO (SNDGQ), distincte de la première et de la deuxième mémoires, et une quatrième mémoire de type FIFO (RCGQ), distincte des trois premières, par le fait que la première unité (DU) contient un registre d'indications de garde (GR), par le fait que l'unité centrale (CU) est apte

à délivrer une instruction gardée au moyen de mémoire (AIDQ) à l'une ou d'une autre unité de traitement (GU) différente de la première, ou une instruction de chargement gardée à la première unité de traitement (DU),  
et

5 à délivrer au moyen de mémoire de la première unité de traitement (DU), une instruction de transmission (SNDGx) destinée à faire transmettre à ladite autre unité (GU) ou à ladite quatrième mémoire (RCGQ), respectivement, la valeur de l'indication de garde (Gx) associée à ladite instruction gardée ou à ladite instruction de chargement gardée,  
10 respectivement,

caractérisé par le fait que les moyens d'aiguillage (DISP) sont aptes à stocker chaque instruction de transmission (SNDGx) dans la troisième mémoire (SNDGQ), à alonger jusqu'à la tête (ETG4) de cette troisième

et par le fait que le processeur comporte en outre des deuxièmes moyens de commande (MCTL) aptes à

15 à extraire de la troisième mémoire (SNDGQ) une instruction de transmission (SNDGx) ayant atteint la tête (ETG4) de cette troisième mémoire, si aucune instruction modificatrice (GMix) temporellement plus ancienne destinée à modifier la valeur de l'indication de garde associée à cette instruction de transmission, n'est présente dans la deuxième mémoire (DIDQ), et à

20 en présence d'une telle instruction modificatrice (GMix) temporellement plus ancienne, à extraire l'instruction de transmission de la troisième mémoire (SNDGQ) seulement après que l'instruction modificatrice (GMix) a été extraite de la deuxième mémoire (DIDQ).

19. Processeur selon la revendication 18, caractérisé par le fait que les deuxièmes moyens de commande (MCTL) comportent

un deuxième compteur de lecture (RDCTR2) incrémenté à chaque fois qu'une instruction est extraite de la deuxième mémoire,

30 un deuxième compteur d'écriture (WRCTR2) incrémenté à chaque fois qu'une instruction est stockée dans la deuxième mémoire,

un ensemble de deuxièmes registres élémentaires (GT0-GT15) respectivement associés à l'ensemble des indications de garde (G0-G15),

40 un cinquième bloc de commande (MC0) apte, à chaque fois qu'une instruction (GMix) modifiant la valeur d'une indication de garde

10 (Gx) est stockée dans la deuxième mémoire, à stocker la valeur courante (WRVL2) du deuxième compteur d'écriture dans un champ principal (CHx) du deuxième registre élémentaire (GTx) associé à cette indication de garde,

15 5. Procédure selon un sixième bloc de commande (MC1) apte à déterminer le caractère toujours présent de cette instruction modificatrice (GMix) dans la deuxième mémoire, et comportant des deuxième moyens de comparaison (MC10) aptes à comparer le contenu du champ principal (CHx) du deuxième registre élémentaire (GTx) avec la valeur courante (RDVL2) du deuxième compteur de lecture.

20 Procédure selon la revendication 19, caractérisé par le fait que le deuxième compteur d'écriture et le deuxième compteur de lecture ont une taille binaire identique égale à la profondeur de la deuxième mémoire (DIDQ), par le fait qu'à chaque deuxième compteur est associé un bit de dépassement (WROVFL2, RDOVFL2) changeant de valeur à 15 chaque fois que le deuxième compteur correspondant revient à sa valeur initiale, par le fait que chaque deuxième registre élémentaire (GTx) comporte en outre un champ auxiliaire à un bit (Bx), par le fait que le cinquième bloc de commande (MC0) est apte, à chaque fois qu'une 20 instruction (GMix) modifiant la valeur d'une indication de garde est stockée dans la deuxième mémoire, à stocker également la valeur courante du bit de dépassement (WROVFL2) du deuxième compteur d'écriture dans le champ auxiliaire (Bx) du deuxième registre élémentaire correspondant, et par le fait que le sixième bloc de commande 25 (MC1) comporte des deuxième moyens de comparaison auxiliaires (PLO1) aptes à comparer la valeur courante du bit de dépassement (RDOVFL2) du deuxième compteur de lecture avec le contenu du champ auxiliaire (Bx).

30 Procédure selon la revendication 20, caractérisé par le fait que les deuxième moyens de comparaison auxiliaires comportent une porte logique NON OU EXCLUSIF (PLO1).

35 Procédure selon l'une des revendications 19 à 21, caractérisé par le fait que chaque étage (ETGi) de la troisième mémoire (SNDGQ) comporte un champ utile (CHUi) pour le stockage d'une instruction de transmission (SNDGx), et un premier champ supplémentaire (Sndgtag),

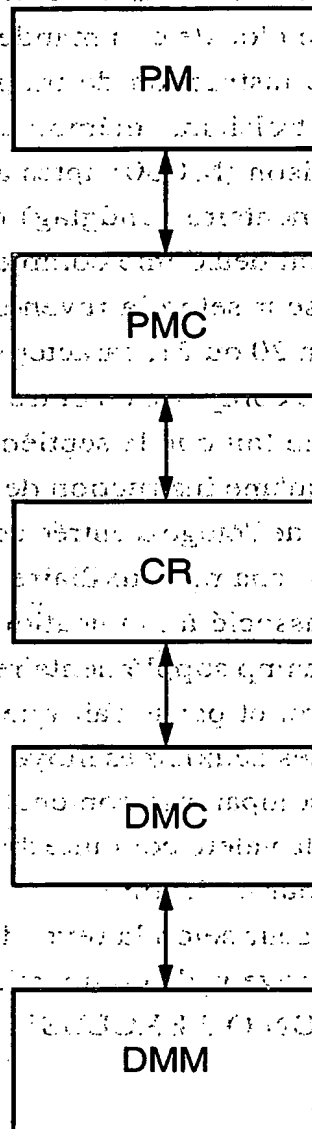
par le fait que les deuxièmes moyens de commande (MCTL) comportent un septième bloc de commande (MC2) apte, à chaque fois qu'une instruction de transmission est stockée dans le champ utile (CHU1) de l'étage d'entrée de la troisième mémoire, à transférer le contenu du champ principal (CHx) du deuxième registre élémentaire (GTx) associé à l'indication de garde correspondante (Gx), dans le premier champ supplémentaire (Sndgtag) de l'étage d'entrée de la troisième mémoire, et

un huitième bloc de commande (MC3) apte à élaborer le critère d'extraction de cette instruction de transmission ayant atteint l'étage de tête (ETG4) de la troisième mémoire, et comportant des deuxièmes moyens de comparaison (MC30) aptes à comparer le contenu dudit premier champ supplémentaire (Sndgtag) de l'étage de tête avec la valeur courante (RDVL2) du deuxième compteur de lecture.

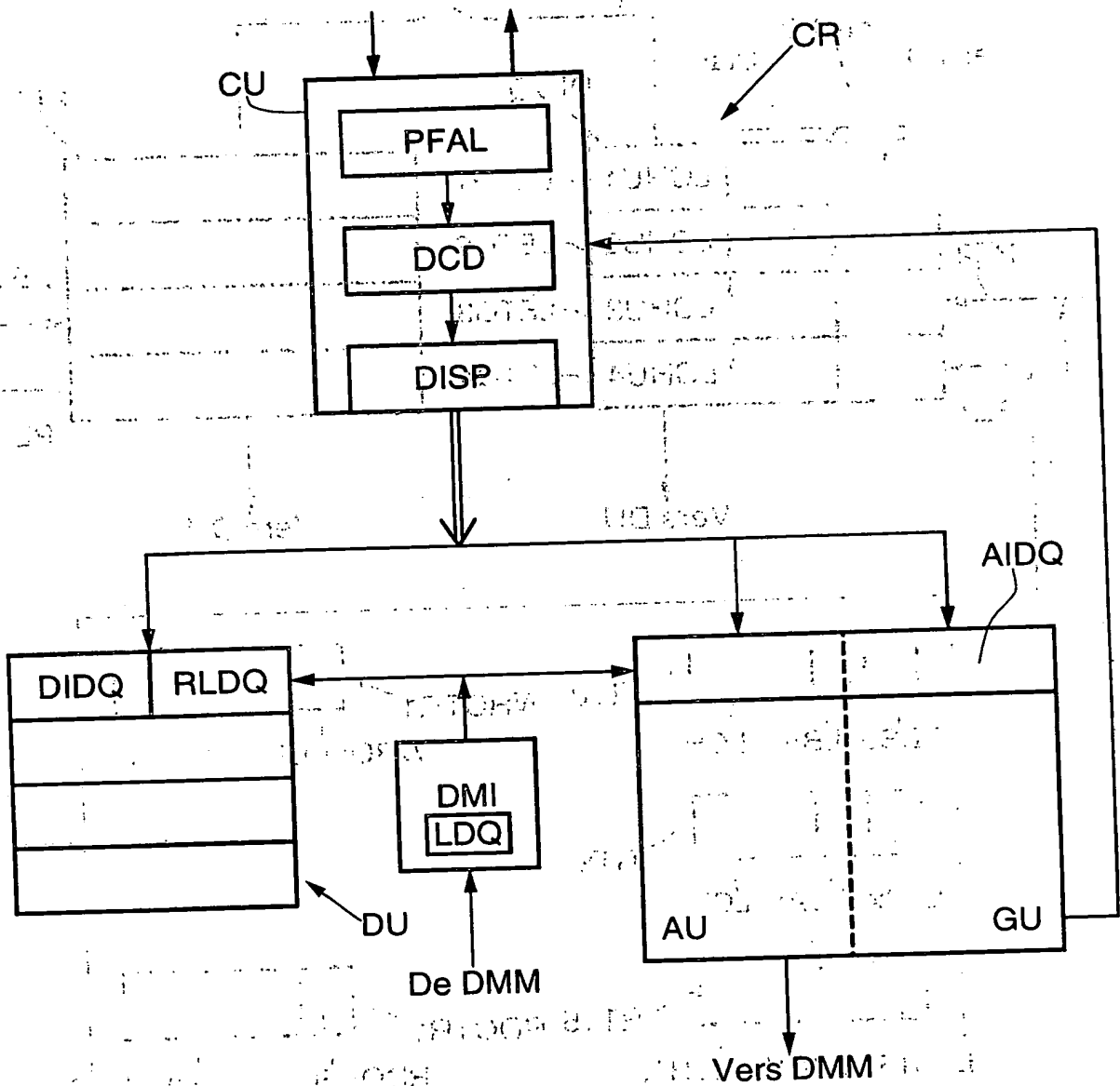
23. Processeur selon la revendication 22 prise en combinaison avec la revendication 20 ou 21, caractérisé par le fait que chaque étage de la troisième mémoire comporte en outre un deuxième champ supplémentaire (Auxgmi), par le fait que le septième bloc de commande (MC2) est apte, à chaque fois qu'une instruction de transmission est stockée dans le champ utile (CHU1) de l'étage d'entrée de la troisième mémoire, à transférer le contenu du champ auxiliaire (BAx) du deuxième registre élémentaire (GTx) associé à l'indication de garde correspondante (Gx), dans le deuxième champ supplémentaire (Auxgmi) de l'étage d'entrée de la troisième mémoire, et par le fait que le huitième bloc de commande (MC3) comportent des deuxièmes moyens de comparaison supplémentaires (PLO3) aptes à comparer le contenu du deuxième champ supplémentaire (Auxgmi) avec la valeur courante du bit de dépassement (RDOVFL2) du deuxième compteur de lecture.

24. Processeur selon la revendication 23, caractérisé par le fait que les deuxièmes moyens de comparaison supplémentaires comportent une porte logique NON OU EXCLUSIF (PLO3).

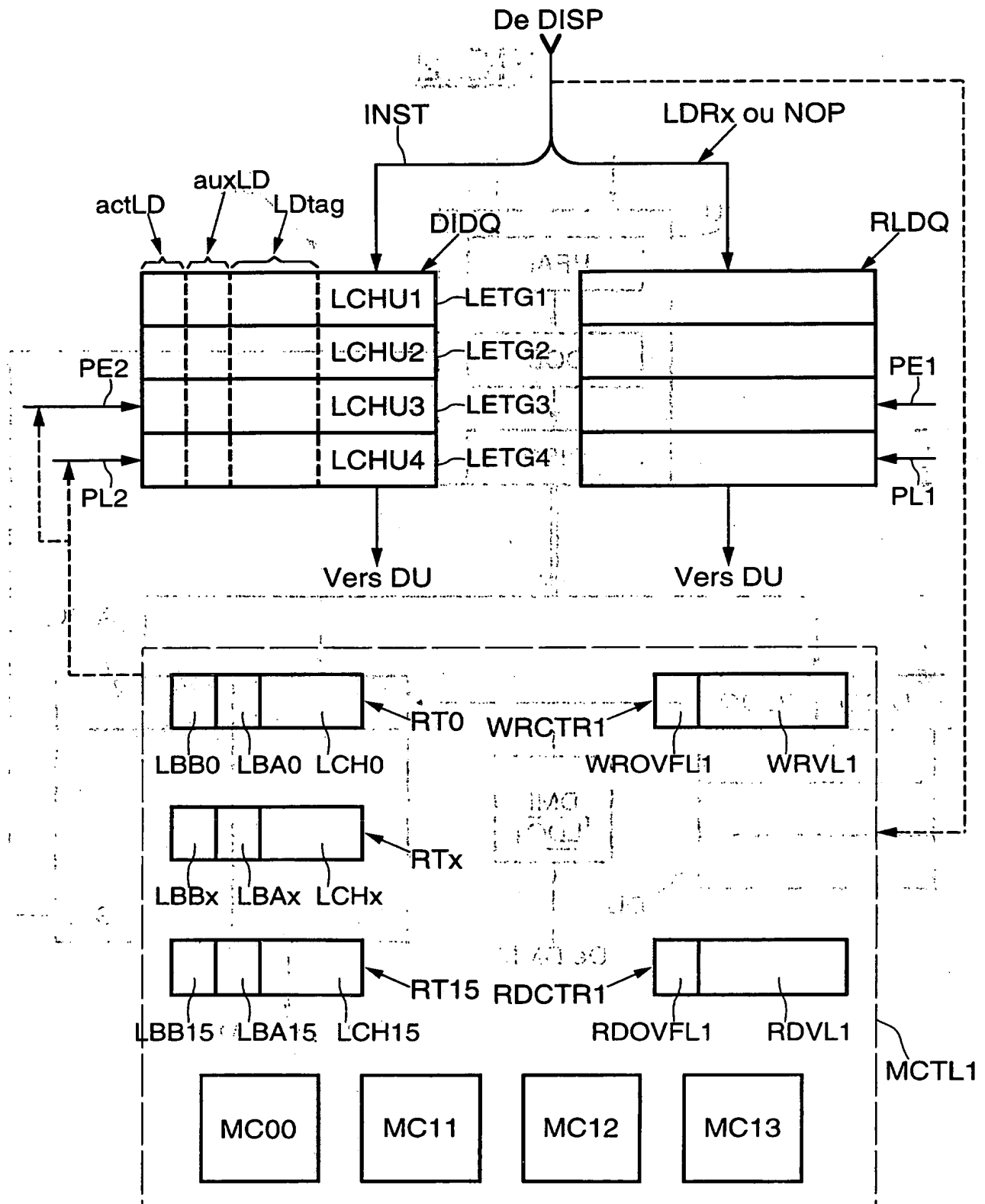


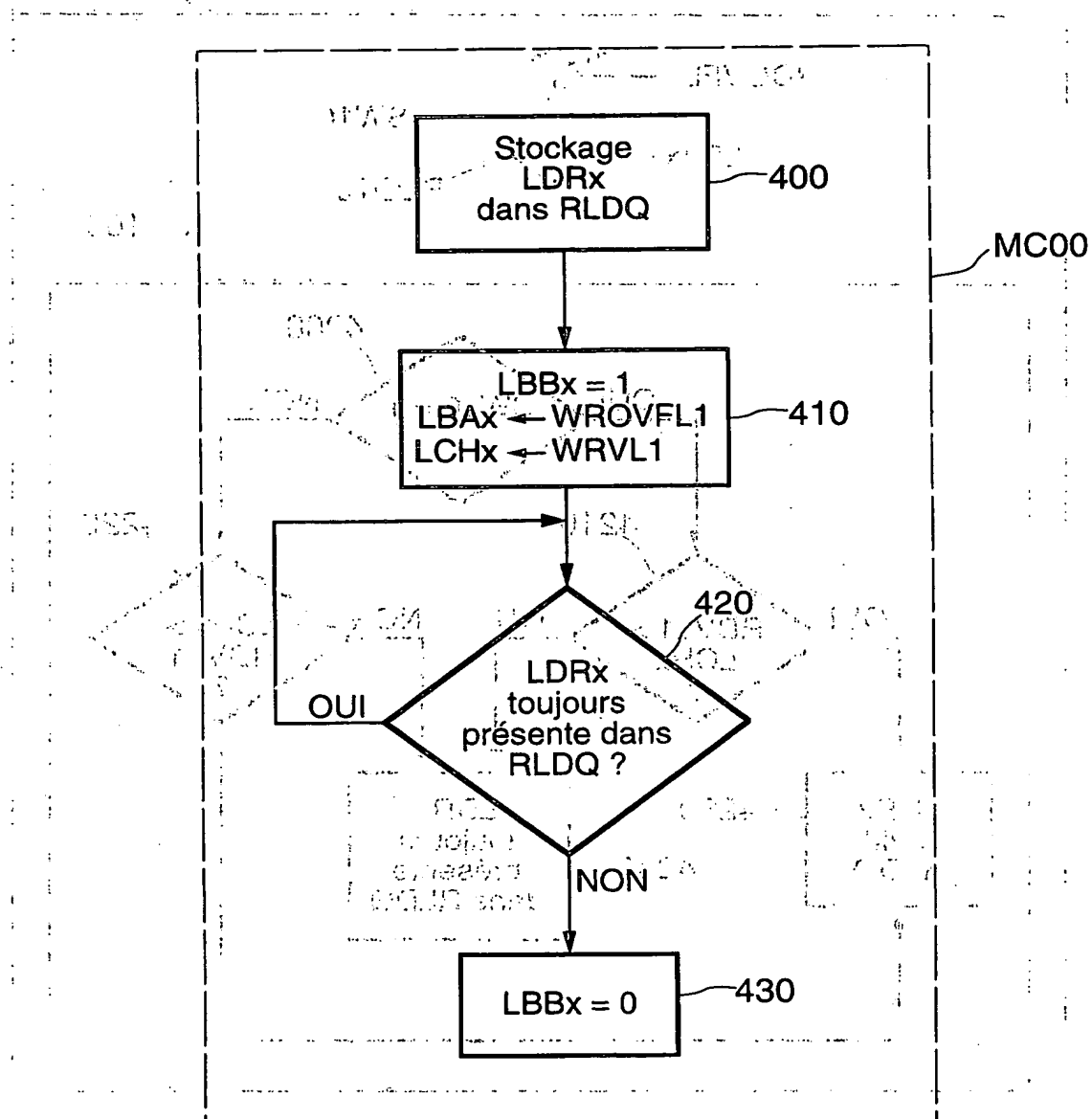
3<sup>e</sup> 1/14**FIG. 1****PROC**

2/14

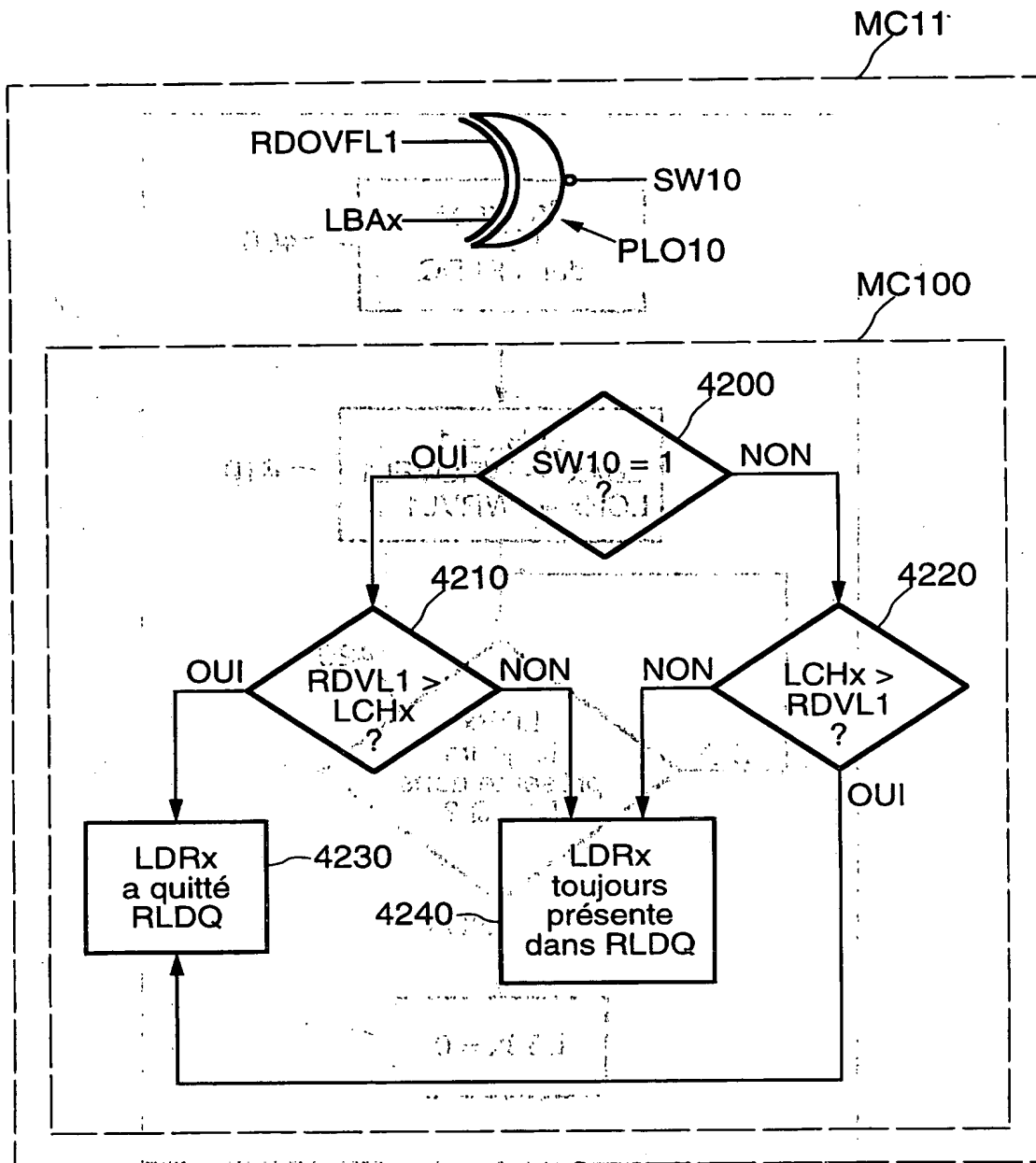
FIG.2

3/14

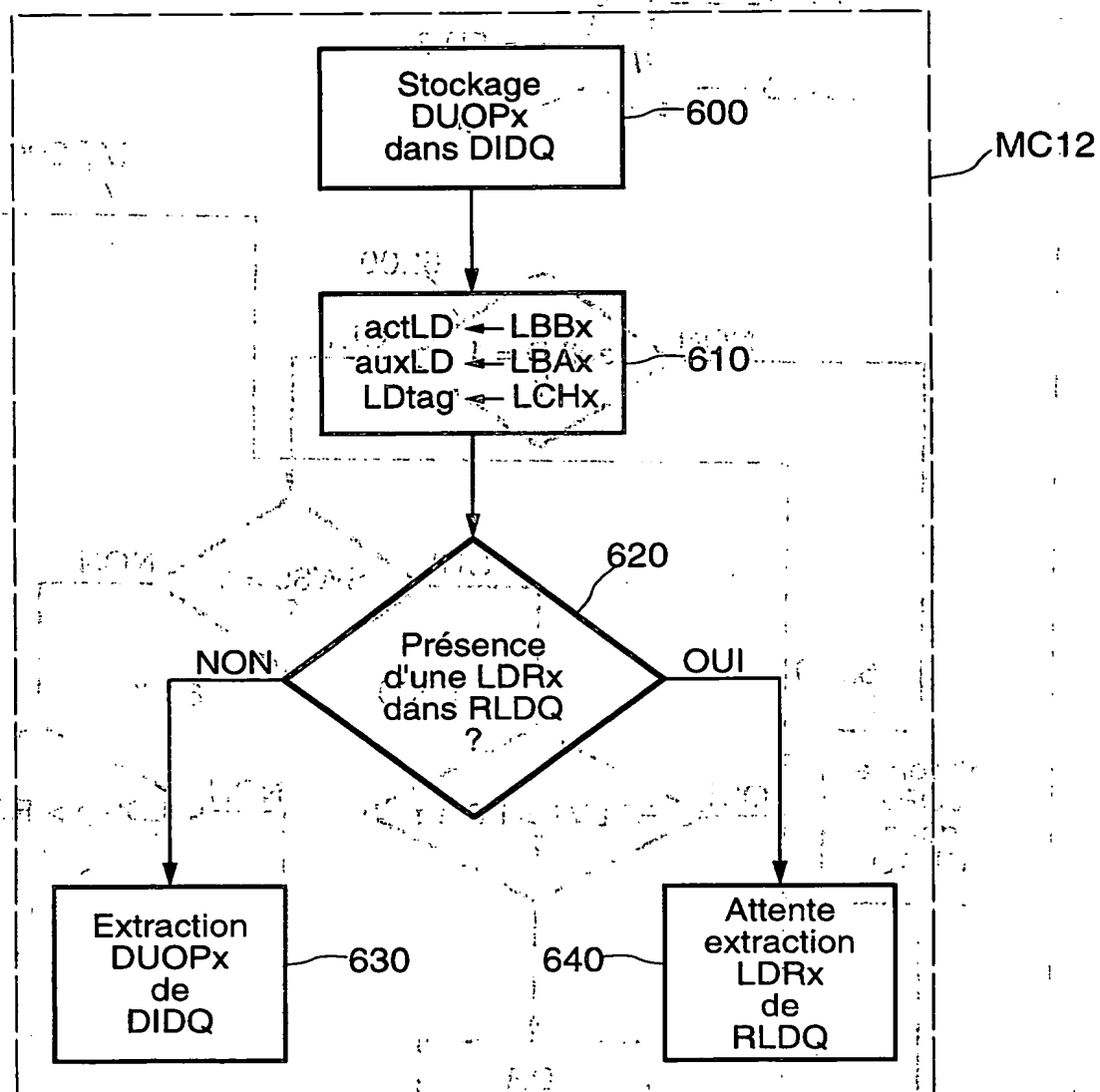
**FIG.3**

**FIG.4**

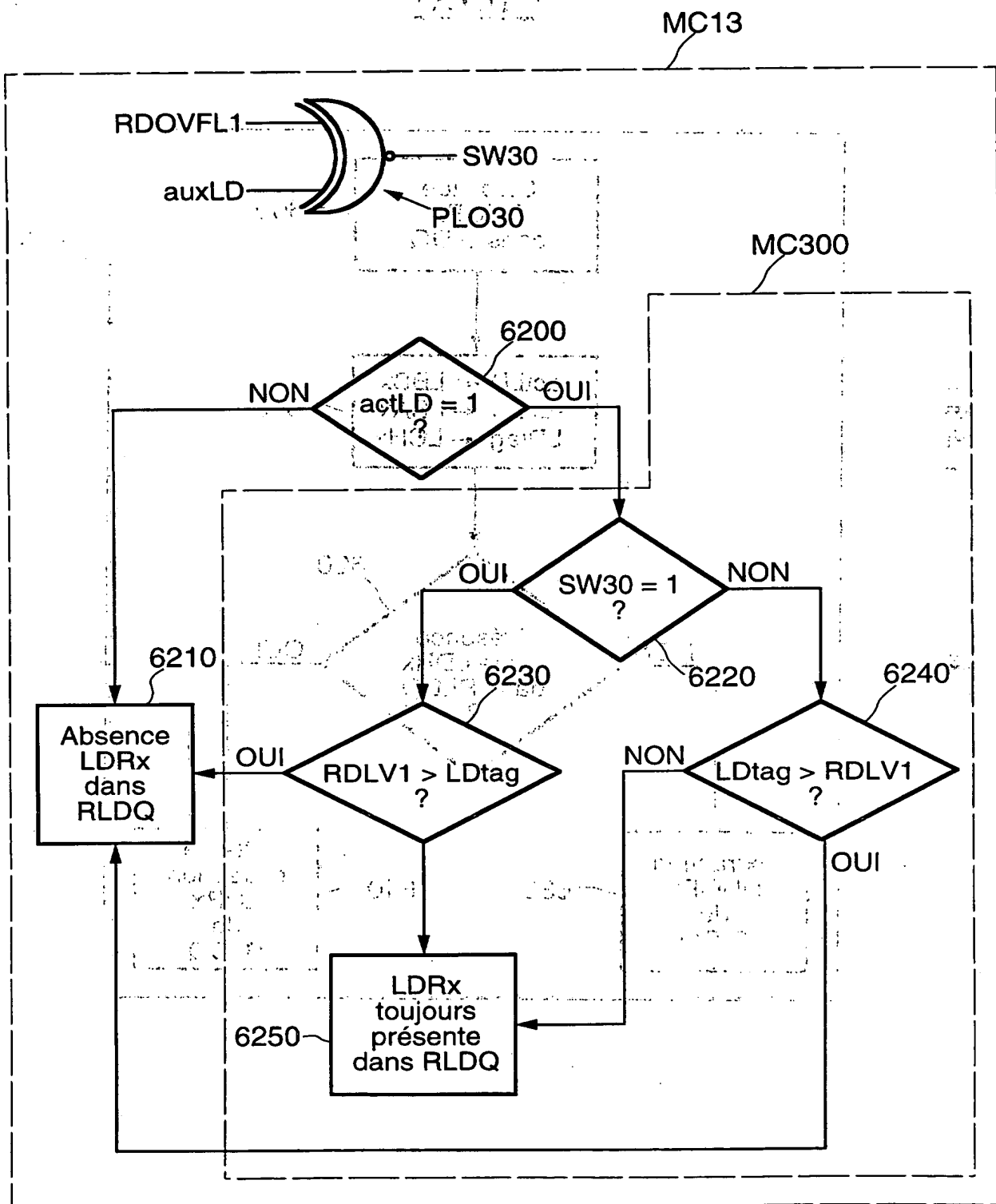
5/14

**FIG.5**

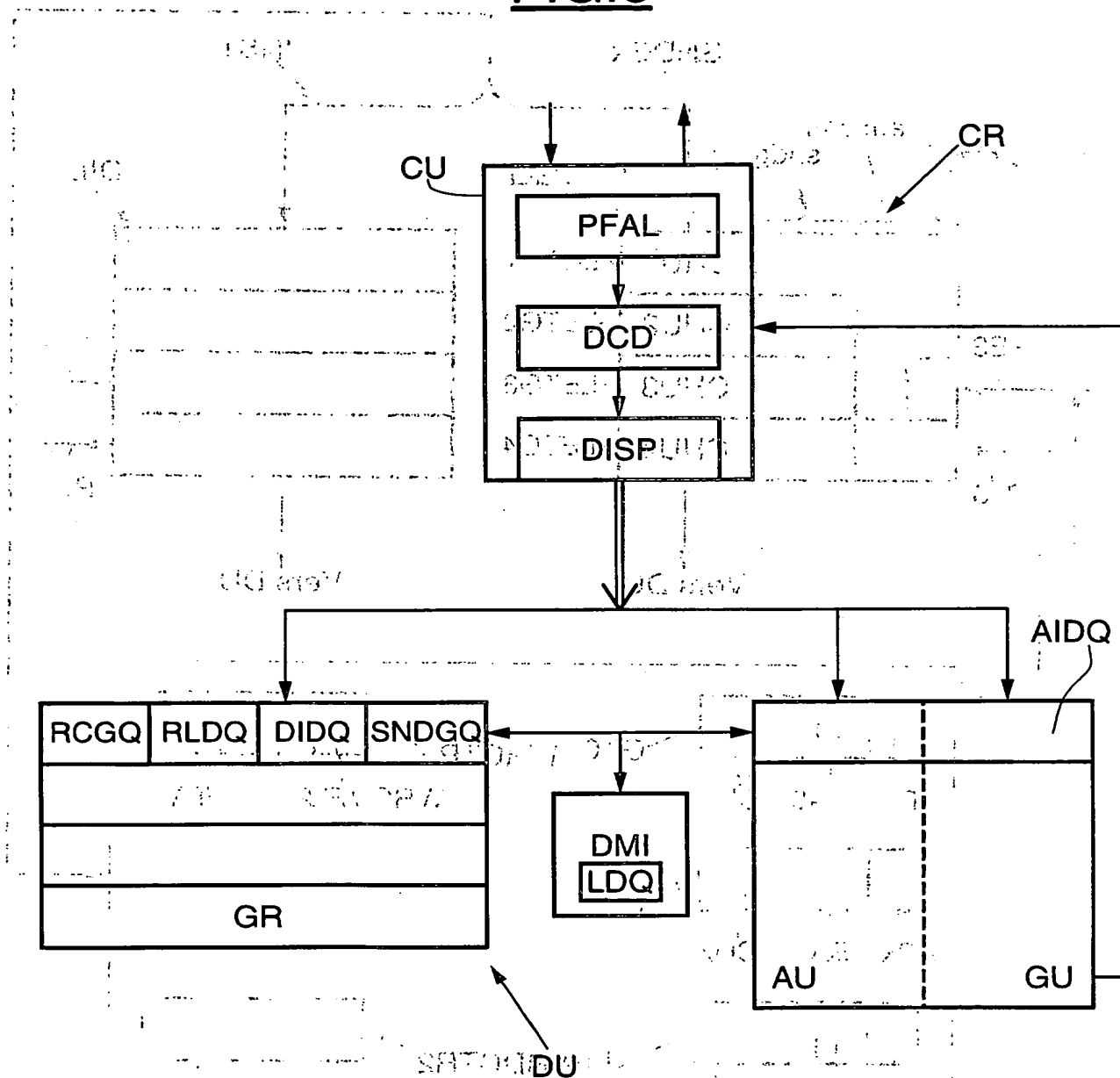
6/14

**FIG.6**

7/14

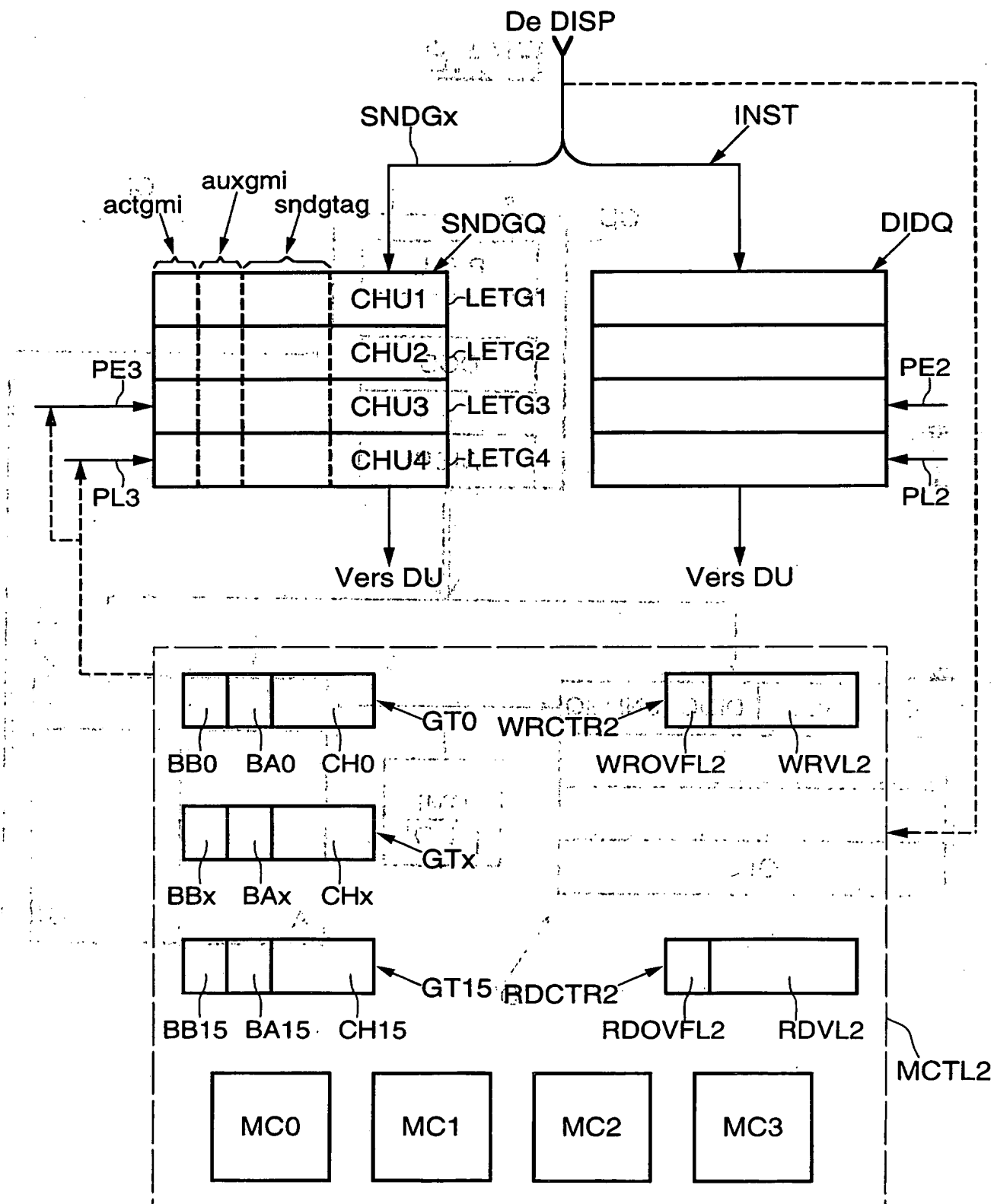
**FIG.7**

8/14

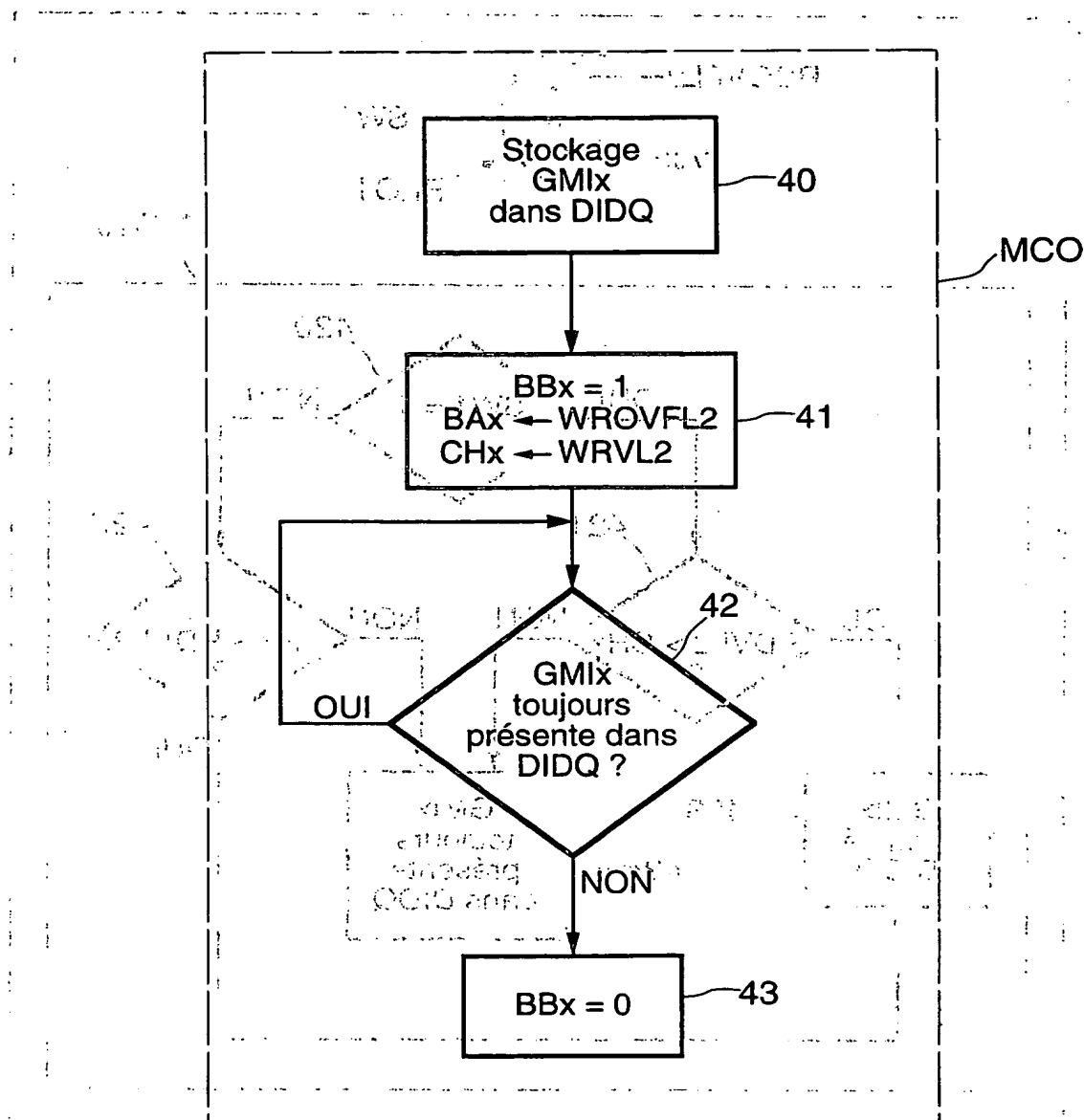
**FIG.8**



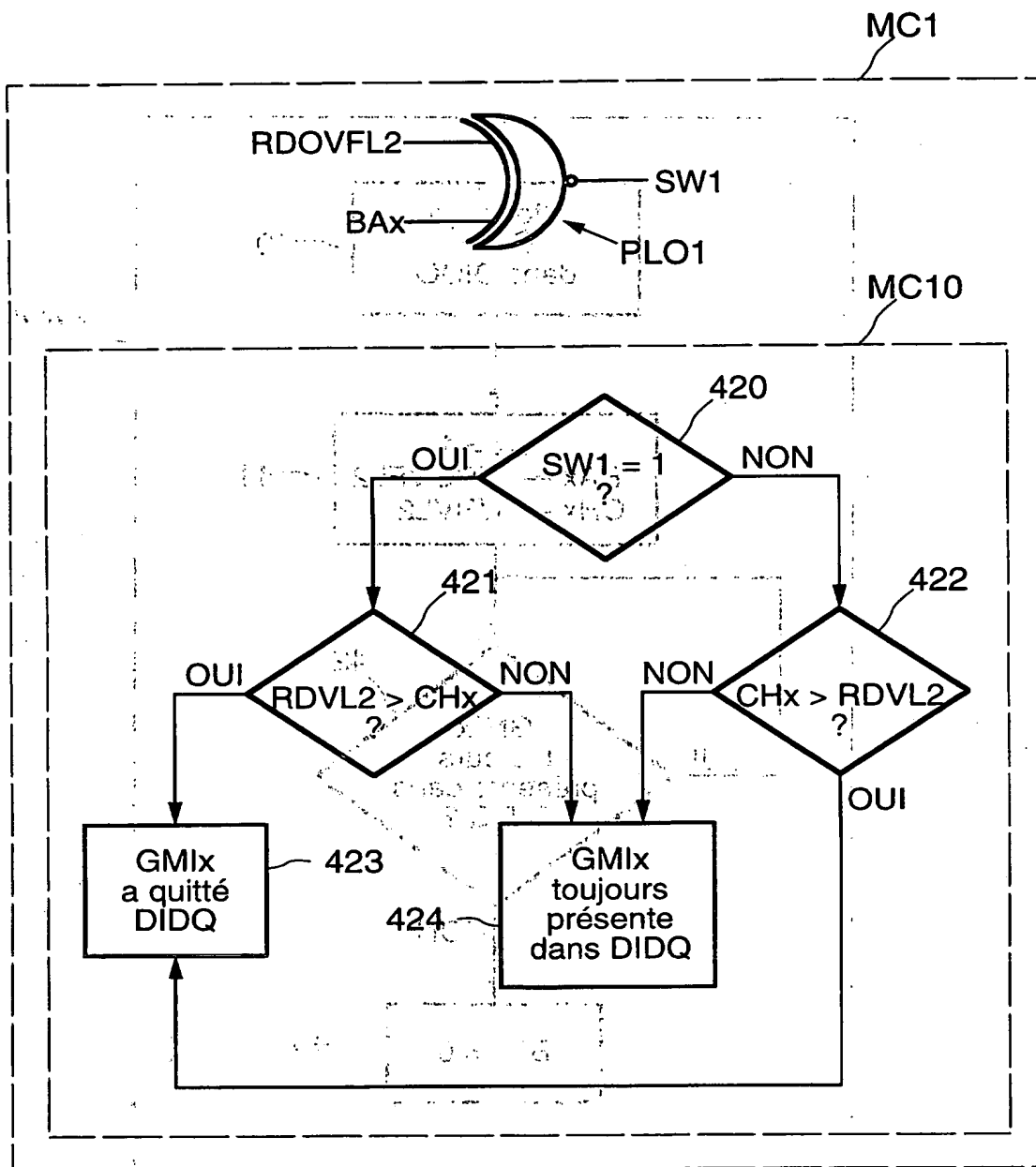
9/14

**FIG.9**

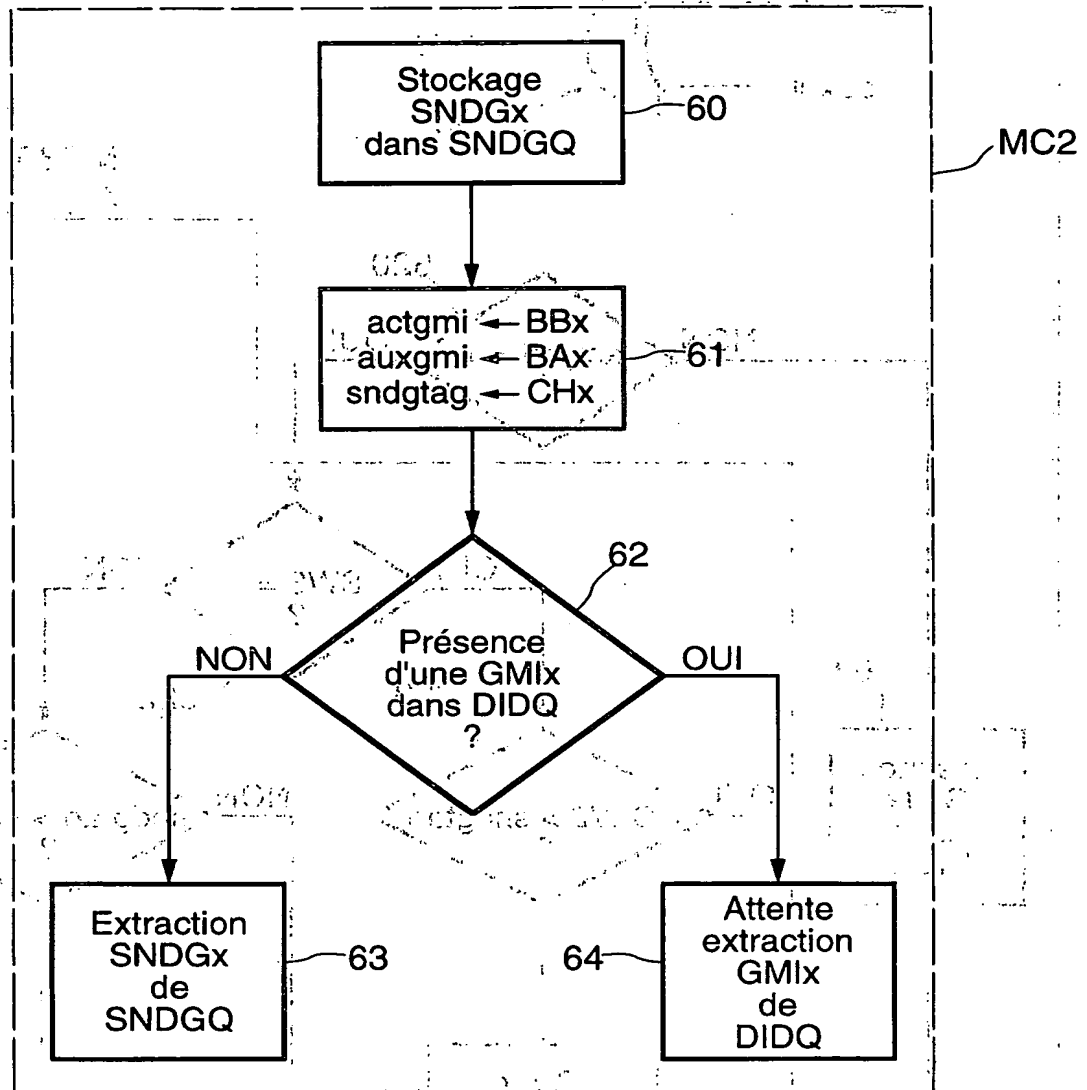
10/14

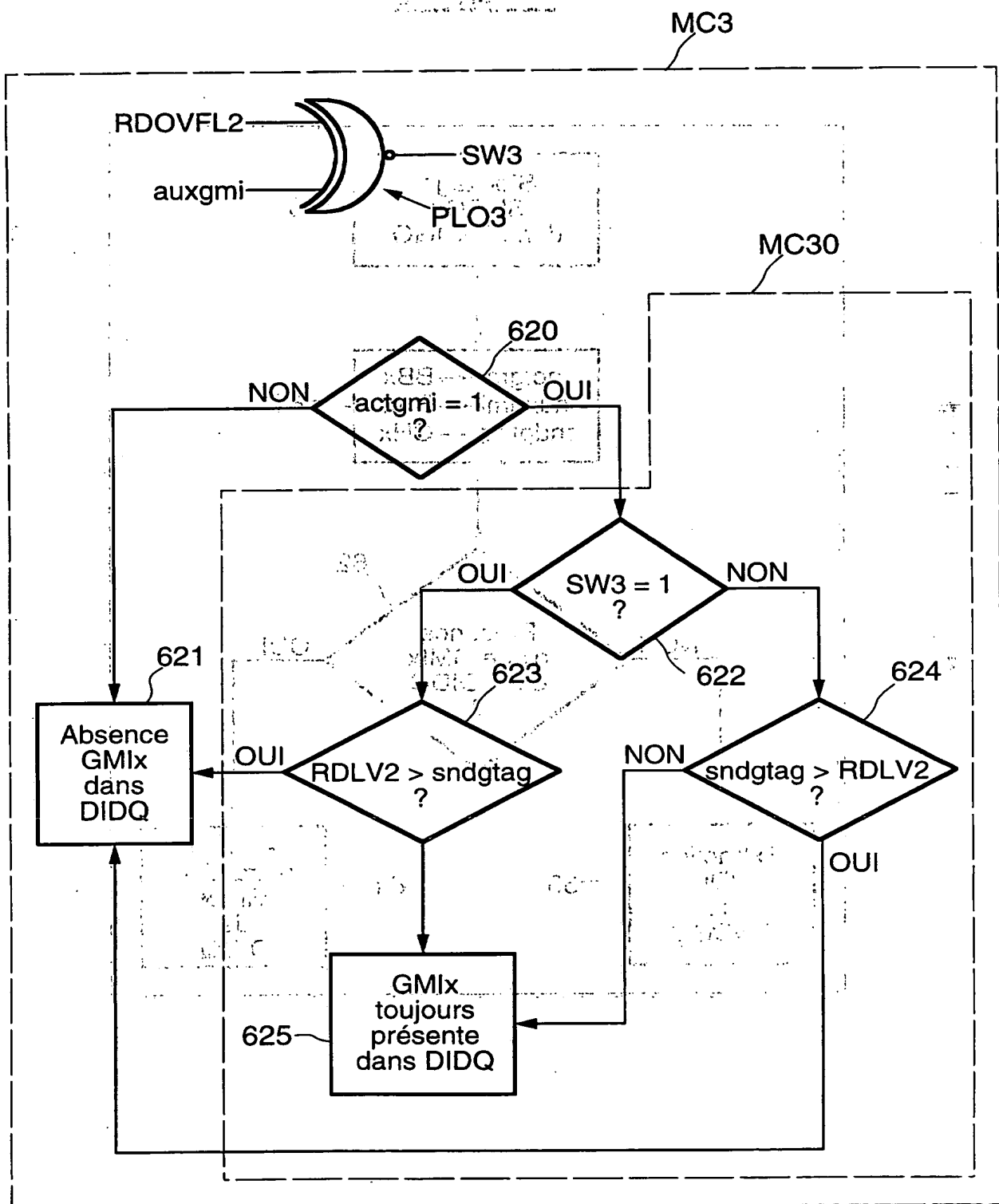
**FIG.10**

11/14

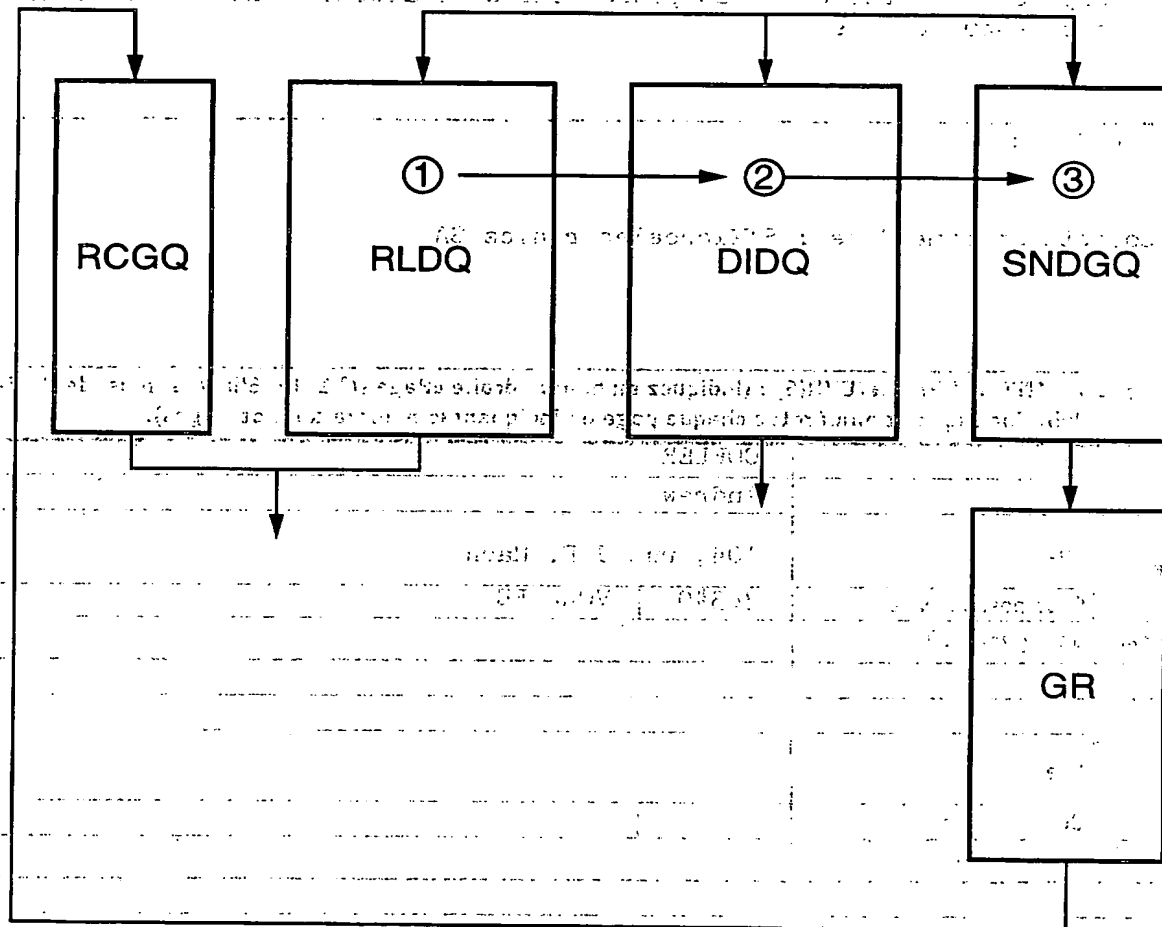
**FIG. 11**

12/14

**FIG.12**

**FIG.13**

14/14

**FIG. 14**

**BREVET D'INVENTION****CERTIFICAT D'UTILITÉ**

Code de la propriété intellectuelle - Livre VI



N° 11235\*02

**DÉPARTEMENT DES BREVETS**

26 bis, rue de Saint Pétersbourg

75800 Paris Cedex 08

Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

**DÉSIGNATION D'INVENTEUR(S)** Page N° .1 / .1.

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

08 113 W / 260899

<b>Vos références pour ce dossier</b> (facultatif)		B 00/4116 FR	
<b>N° D'ENREGISTREMENT NATIONAL</b>		0102647	
<b>TITRE DE L'INVENTION</b> (200 caractères ou espaces maximum)			
Procédé de gestion d'instructions au sein d'un processeur à architecture découplée, en particulier un processeur de traitement numérique du signal, et processeur correspondant.			
<b>LE(S) DEMANDEUR(S) :</b>			
Société Anonyme dite : STMicroelectronics SA			
<b>DESIGNE(NT) EN TANT QU'INVENTEUR(S) :</b> (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
<b>Nom</b>		COFLER	
<b>Prénoms</b>		Andrew	
<b>Adresse</b>	<b>Rue</b>	104, rue J.F. Hache	
	<b>Code postal et ville</b>	38340	VOREPPE
<b>Société d'appartenance (facultatif)</b>			
<b>Nom</b>			
<b>Prénoms</b>			
<b>Adresse</b>	<b>Rue</b>		
	<b>Code postal et ville</b>		
<b>Société d'appartenance (facultatif)</b>			
<b>Nom</b>			
<b>Prénoms</b>			
<b>Adresse</b>	<b>Rue</b>		
	<b>Code postal et ville</b>		
<b>Société d'appartenance (facultatif)</b>			
<b>DATE ET SIGNATURE(S)</b> <b>DU (DES) DEMANDEUR(S)</b> <b>OU DU MANDATAIRE</b> (Nom et qualité du signataire)		Paris, le 27 Février 2001.	
		 A. CASALONGA (bm 92-1044i) Conseil en Propriété Industrielle	

**THIS PAGE BLANK (USPTO)**